MISSION-BASED GUIDANCE SYSTEM DESIGN FOR AUTONOMOUS UAVS

A Thesis
Presented to
The Academic Faculty

by

Jongki Moon

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2009

MISSION-BASED GUIDANCE SYSTEM DESIGN FOR AUTONOMOUS UAVS

Approved by:

Dr. J. V. R. Prasad, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Mark Costello
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Daniel P. Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Patricio A. Vela
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Eric N. Johnson
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: September 28, 2009

To my family

사랑하는 나의 가족들에게

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. J.V.R. Prasad for providing me with the great opportunity to study and work on my research. I deeply appreciate him for his insightful academic advice, continuous support, and encouragement. I also thank him for being patient with my research progress. I would like to thank Dr. Daniel Schrage, Dr. Eric Johnson, Dr. Mark Costello, and Dr. Patricio Vela for serving on my thesis committee, and for giving me helpful suggestions and comments. I also thank the other faculties at Georgia Tech who offered me great lessons.

I would like to express my gratitude to people I met during my Ph.D. work at Georgia Tech. In particular, I would like to thank Dr. Suresh Kannan, Dr. Ramachandra Sattigeri, and Dr. Suraj Unnikrishnan for helping me with implementation of my algorithms onto the flight system. I would like to thank my friends, Dr. Chang Chen, Dr. James Rigsby, Ivan Grill, Fahri Ersel Olcer, and An Binh Vu. I would like to thank all my Korean friends for their support and encouragement. I would like to express my special thanks to Keeryun Kang for his help and advice.

I would like to give my sincere appreciation to my parents, parents-in-law, and brothers. Without their support and pray, I could not have accomplished my Ph.D. degree. Their endless love encouraged me to go through all the difficulties during my study. Finally, I would like to thank my wife and son for their love, support, and patience. They are a real reason for having this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The advantages of UAVs in the aviation arena have led to extensive research activities on autonomous technology of UAVs to achieve specific mission objectives. This thesis mainly focuses on the development of a mission-based guidance system. Among various missions expected for future needs, autonomous formation flight (AFF) and obstacle avoidance within safe operation limits are investigated.

In the design of an adaptive guidance system for AFF, the leader information except position is assumed to be unknown to a follower. Thus, the only measured information related to the leader is the line-of-sight (LOS) range and angle. Adding an adaptive element with neural networks into the guidance system provides a capability to effectively handle leader's velocity changes. Therefore, this method can be applied to the AFF control systems that use a passive sensing method. In this thesis, an adaptive velocity command guidance system and an adaptive acceleration command guidance system are developed and presented. Since relative degrees of the LOS range and angle are different depending on the outputs from the guidance system, the architecture of the guidance system changes accordingly.

Simulations and flight tests are performed using the Georgia Tech UAV helicopter, the GTMax, to evaluate the proposed guidance systems. The simulation results show that the neural network (NN) based adaptive element can improve the tracking performance by effectively compensating for the effect of unknown dynamics. It has also been shown that the combination of an adaptive velocity command guidance

system and the existing GTMax autopilot controller performs better than the combination of an adaptive acceleration command guidance system and the GTMax autopilot controller. The successful flight evaluation using an adaptive velocity command guidance system clearly shows that the adaptive guidance control system is a promising solution for autonomous formation flight of UAVs.

In addition, an integrated approach is proposed to resolve the conflict between aggressive maneuvering needed for obstacle avoidance and the constrained maneuvering needed for envelope protection. A time-optimal problem with obstacle and envelope constraints is used for an integrated approach for obstacle avoidance and envelope protection. The Nonlinear trajectory generator (NTG) is used as a real-time optimization solver. The computational complexity arising from the obstacle constraints is reduced by converting the obstacle constraints into a safe waypoint constraint along with an implicit requirement that the horizontal velocity during the avoidance maneuver must be non-negative. The issue of when to initiate a time-optimal avoidance maneuver is addressed by including a requirement that the vehicle must maintain its original flight path to the maximum extent possible.

The simulation evaluations are preformed for the nominal case, the unsafe avoidance solution case, the multiple safe waypoint case, and the unidentified obstacle size case. Artificial values for the load factor limit and the longitudinal flap angle limit are imposed as safe operational boundaries. Also, simulation results for different limit values and different initial flight speed are compared. Simulation results using a nonlinear model of a rotary wing UAV demonstrate the feasibility of the proposed approach for obstacle avoidance with envelope protection.

CHAPTER 1

INTRODUCTION

The objective of this thesis is to develop a guidance system which enables autonomous unmanned aerial vehicles (UAVs) to perform a given mission in effective and safe way. It is expected that UAVs are used for various operation environments and their missions vary accordingly. The first mission considered in this thesis is leader-follower formation flight, and the second is obstacle avoidance while maintaining vehicle's maneuver within safe operational boundaries. This requires a vehicle's flight system to have the capabilities of communication, leader and own-ship state estimation, data fusion, and guidance and flight control. Each of these is substantial research topics in their own right. This thesis, however, focuses on the problems of an autonomous guidance system.

1.1 Autonomous Technology for UAVs

UAVs have been referred to in many ways: remotely piloted vehicles (RPVs), drones, robot planes, and pilotless aircraft are a few such names. Most often called UAVs, they are defined as powered, aerial vehicles that do not carry a human operator, use aerodynamics forces to provide vehicle lift, can fly autonomously, or be piloted remotely, can be expendable or recoverable, and can carry a lethal or nonlethal payload [6]. UAVs range in size from a few inches to hundreds of feet, can be fixed or rotary wing aircraft, can be remotely piloted or autonomous. UAVs are either descried as a single air vehicle, or a UAV system, which usually consists of a group of air vehicles, a ground control

station, and support equipment. UAVs are expected to play an important role in both civil and military applications. A wide range of applications includes the following:

- weather and atmospheric research
- surveillance and reconnaissance
- conventional combat roles
- innovative roles that were not previously possible (for example, "dull, dirty, and dangerous" missions, such as operations in chemical and biological hazardous environments and operations that require micro air vehicles)

A principal reason for the interest in UAVs is the desire to reduce the risk to humans, but it also is to perform missions in a more efficient and less costly fashion than has historically been the case with manned vehicles. A related reason is that freeing machines from the limitations imposed by humans would increase their performance. The hope has been that unmanned air vehicles would be less expensive to develop and manufacture than manned aircraft, and that UAVs will reduce the demand for the supporting facilities and manpower that modern aircraft require.

The advantages of UAVs in the aviation arena have led to extensive research activities on autonomous technology of UAVs to achieve specific mission objectives. In fact, as a result of technological advances in flight control, data and signal processing, off-board sensors, communications links, and integrated avionics, UAVs can be now serious assets. The single fundamental feature that most distinguishes UAVs from other aerial vehicles is autonomy, or "built-in intelligence." [6][70][99] Autonomy can be defined as the capability for UAVs to make decision, or the degree of self-reliance and independence [6]. Autonomous control enables many functions, normally accomplished by humans, to be done by the vehicle instead. The Autonomous Control Level (ACL) was proposed to measure attributes of the autonomy shown in Figure 1 [15]. Since this metric is based on how humans make decisions, it can provide finer measurement at what level

UAV decisions are made, and to target specific gaps in technology. Technically, the autonomy for UAVs can be achieved from three layers of functional hierarchy. The first and lower layer consists of individual vehicle's flight control functions; the second consists of vehicle-management functions; the third and highest layer consists of mission-management functions. Even though only the flight control and vehicle-management layers are implemented onboard each vehicle in the current UAV systems, the mission-management layer is expected to be implemented onboard in the near future.

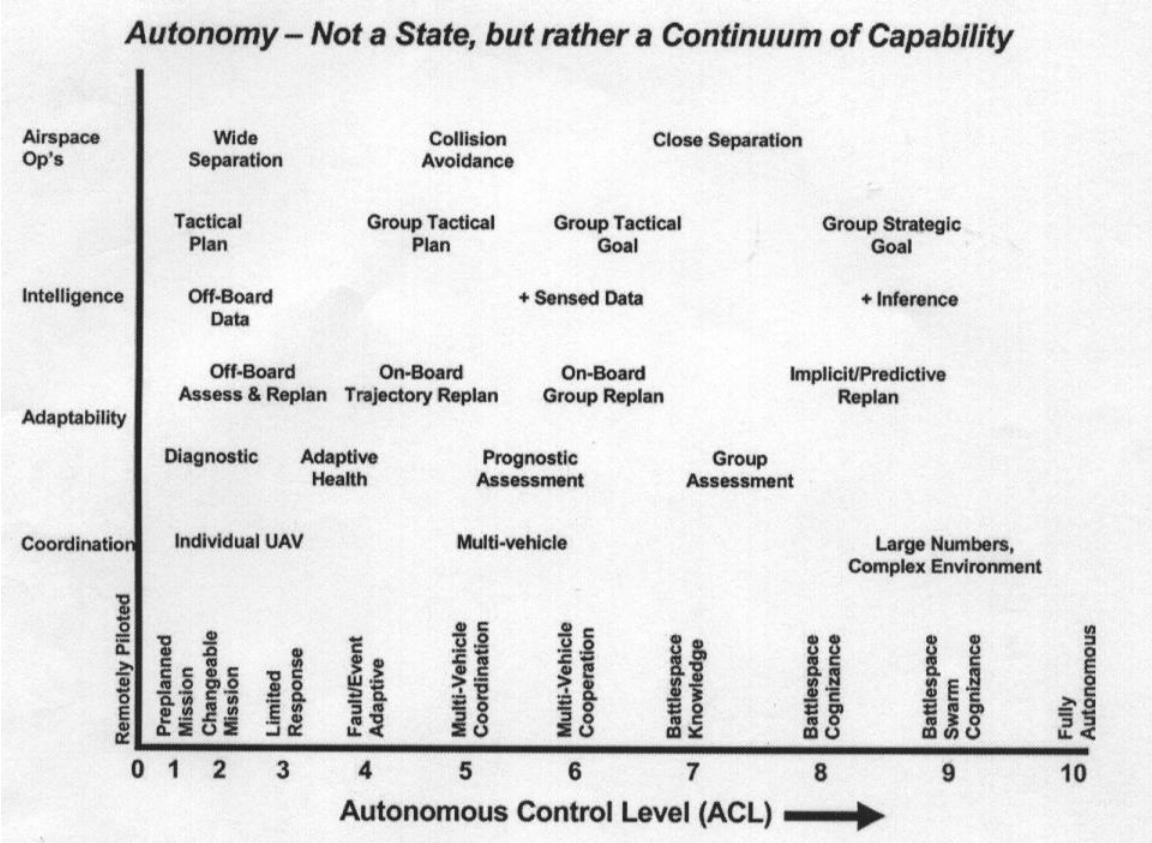


Figure 1. Machine decision capability versus ACL level [15]

The function of the flight control layer is to ensure vehicle stability, to maintain desired flight parameters, and control the vehicle's subsystems in response to the vehicle-management layer. Common control modes are acceleration/rate command, maintain

altitude/speed/heading, automatic landing/take-off, and trajectory following. The current state-of-art of control design for UAVs includes the basic techniques of multivariable robust control theory for linearized systems, feedback linearization/dynamic inversion for nonlinear systems, and so on. Although these techniques are also applied for conventional manned aircraft, the nature of UAVs changes the design problem. The absence of onboard manual control eliminates the requirement related to the handling quality and pilot comfort. The function of vehicle-management layer is to manage onboard vehicle operations, which include planning the vehicle's mission time line, establishing proper operating modes/tactics, and monitoring vehicle health. The function of the mission-management layer is to plan, rehearse, and assign missions to collection of vehicles. This layer is responsible for effective management of a group of UAVs.

1.2 Autonomous Formation Flight

Recently, the formation control of multi-agent systems has attracted many researchers from diverse fields in scientific and engineering disciplines. The objective of formation control is to drive a group of agents to move together in desired formation and accomplish desired tasks. The idea of formation by design is inspired by observation of many examples in biology, such as swarming, bird flocking, and fish schooling [71]. The subject of autonomous formation flight (AFF) has also become one of the most interesting fields in the aerospace community. AFF is very important especially for UAVs since it is the most representative example of the multi-vehicle coordination/cooperation missions.

Management of the UAV formation can be centralized or decentralized. In the former case, one formation manager acts as a supervisor for all vehicles and manages the topology of the channels used to exchange information among them. This manager can be one of the UAVs or ground based. The centralized approach has several disadvantages:

- the amount of communications required among all entities may be undesirable
- mission requirements may not allow continuous communications
- formation failure must be detected and recovered as rapidly as possible

In a decentralized management scheme, on the other hand, each UAV is given a certain level of decision capability, while the whole formation must be capable of reconfiguring, making decision, and achieving mission goals. The advantages of distributing the management are several:

- only inter-vehicle information must be exchanged
- non-radio based communication such as optical sensors can be used because of the very small distance among vehicles. This is very important for military applications.
- reaction times can be minimized.

The various approaches considered for AFF over past decades include Proportional-Integral-Derivative (PID) control [77], Linear Quadratic Regulator [105], and nonlinear adaptive control [87]. Formation control is also considered as coordinated motion of a group of vehicles in the mobile ground robotics community [1][85][87][95]. Almost all of control approaches described above assume that measurements of the leader position and velocity, and angular attitudes are available for feedback controller. It is also assumed that each vehicle is equipped with navigation systems to obtain its own position and velocity information. As a result, at least one radio channel is required to transmit leader information to the follower. In such case, a fault in the receiver or transmitter may cause significant damage to mission success. Specially, when active communication is not allowed for the purpose of the radio silence, this type of approach has a limitation for military applications. Therefore, passive detection methods are preferred rather than active communication. Examples of passive sensing include detection of wakes generated by the leader [73] and use of vision sensors onboard the vehicle [42][104]. Since passive

detection methods can measure only some of relative motion information, Extended Kalman Filter (EKF) to estimate necessary leader information or Adaptive control component to compensate for uncertainty in relative motion is required.

1.3 Obstacle Avoidance for Autonomous Vehicles

This research mainly focuses on the technique for the vehicle-management layer, especially, technique for trajectory planning and health monitoring. Since UAVs are expected to operate in unknown and adversarial environments, situation awareness and flight mode decision are one of the most essential parts in the vehicle-management layer. Situation awareness requires algorithms that extract objects of interest from the information provided by active or passive sensors, while motion planning needs logics that determine specific flight phases according to given situation and mission. The mission environment may contain obstacles and zones that the vehicle is not allowed to enter and may not be fully characterized at the start of a mission. Obstacles may be detected as the vehicle moves through the environment or their location may change over time. There have been numerous researches on online trajectory planning for obstacle avoidance. The various approaches to obstacle avoidance problem can be categorized as heuristic methods and optimization based methods.

The heuristic method strives to approximate and simplify as many elements of route planning problem as possible. The performance of heuristic approaches, for example, runtime and the suitability of the results, is hard to quantify, thus only able to be measured through simulations and experimental tests. In addition, the results provided by these algorithms may be far from optimal. The heuristic approaches, however, are usually easy to implement and computationally efficient. One of the early stages of the heuristic approaches is a potential function method [45][47][52]. With this method, obstacles are modeled as repulsive potential functions, while target points are replaced by attractive

potential functions [45][47]. Then the resultant force is used as the references for trajectory commands. As mentioned in Reference [52], however, this method has inherent limitations such as trap situation due to local minima and lack of robustness.

Another method of the heuristic approach is to use the visibility graph [57]. By using the robot's configuration space, the problem of planning the path of the robot was reduced to planning the path of a point representing the robot. The visibility graph shows which vertices of the obstacles are visible from other vertices. A safe path is determined by connecting together several safe path segments. A vector based approach [102] is to combine the visibility graph and potential functions. Reactive obstacle avoidance methods also fall into the heuristic approach. Reactive methods generate velocity command [62][63] and/or accelerate command [2] by comparing a current situation with certain criteria, which are determined according to vehicle characteristics and operating environments. In a word, reactive methods can be presented as a series of 'if-then' statements. Flight test results using the heuristic approaches are shown in References [14] [17][23][35].

Besides the heuristic approach, the optimization-based approach is also a main stream of obstacle avoidance. These approaches are attractive because they can theoretically provide optimal performance. This performance, however, comes at a cost with complex problem setups and significant computational requirements, which can be a challenge for real-time applications. Translating the constraints on obstacles, vehicle dynamics, etc., into mathematical presentation is difficult and may produce hundreds of constraints. For example, the standard indirect, direct, and homotopy methods for the numerical solution of optimal control problems have been used in References [13][101]. Formulating time-optimal problem for a constant speed with the actuator limit is shown in References [59][93][94]. Minimum effort guidance for obstacle avoidance is proposed in References [27][104] using collision cone approach suggested in Reference [11]. The

minimum effort guidance is based on a well-known proportional navigation while safe aiming points are determined by the geometric rule.

As mentioned earlier, classical optimal approaches suffer from the computational complexity. Reference [78] investigates the computational complexity of the planning the motion of a vehicle in 2-dimensional or 3-dimensional space. In order to reduce the computational cost and to increase a chance for real-time implementation, several studies on the programming techniques for optimal problems have been shown. One of these modern approaches to real-time optimal problems is to use differential geometric techniques [12][21]. These methods depend on finding special outputs which are called flat outputs. If a system is differentially flat, then the complete differential behavior of the system can be found in terms of outputs and their derivatives. Once the outputs of interest are parameterized by B-splines [7], sequential quadratic programming is used to find the coefficients of the B-splines to satisfy the optimization objectives as well as the constraints. The method is called as the nonlinear trajectory generator (NTG) [60][72]. Following works for planning obstacle-free trajectory are presented in References [20] [37][61].

Another method for solving real-time optimal problems is the mixed-integer linear programming (MILP) [22]. MILP is an optimization framework that allows inclusion of integer variables and discrete logic in a continuous linear optimization problem. These variables can be used to represent logical constraints such as obstacles and collision avoidance rules, while the dynamic properties of the vehicle are formulated as continuous constraints [18][79]. Although MILP can systematically handle obstacle and collision constraints, it cannot handle the computational complexity of a typical planning problem. Online optimization of a less general problem for specific initial states is therefore of most interest. Recent studies [19][82] show that a pseudospectral (PS) method can be an alternative approach to solving nonlinear optimal control problem in real-time. PS methods were originally developed for solving partial differential equations.

Then, during the 1990s, PS methods were introduced for solving optimal control problems and have gained considerable attention. One of the main reasons for the popularity of PS methods is that they offer an exponential convergence rate for the approximation of analytical functions while providing Eulerian-like simplicity. Thus, PS methods generate a smaller scale optimization problem when compared to other methods. Using PS methods, a software package for the real-time optimization problem is recently developed [81]. Flight test results using optimization-based approaches are also shown in References [86][103].

1.4 Flight Envelope Protection

Another important task of the vehicle-management layer in autonomous UAVs is to monitor their health and prevent damage to the vehicle. The UAV health condition depends on aerodynamic loading on the UAV, actuator operating status, structural fatigue, and so on. A technique to maintain these specific flight parameters within the operating envelope is called as envelope protection [30]. This technique is originally developed for manned vehicles because the pilot always has to pay attention to the descried operating limit and the safe guideline is usually selected conservatively. Thus the main purpose of the envelope protection is to reduce a pilot's workload and to extend a safe operating envelope [34]. Figure 2 shows a fundamental idea for envelope protection.

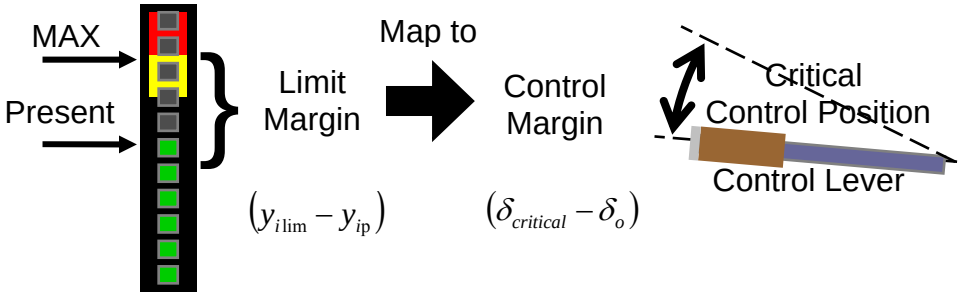


Figure 2. Basic idea for envelope protection

Envelope protection is also important for UAVs. Since maneuverability constraints for the comfort of a human pilot are no longer applicable to UAVs, controller design can be focused solely on meeting mission needs. In addition, UAVs will be operated more aggressively than their manned counterparts, closer to authority limits of actuation and closer to the physical limits of airframes. However, the task of envelope protection must be done automatically due to the absence of a pilot.

An envelope protection system consists of two fundamental functions: limit detection and limit avoidance. One possible approach is to use direct sensor measurement, and then intervene control system to prevent limit violation [50]. This approach has been shown to effectively prevent envelope violations in the sense that it is transparent to the pilot. Another approach to limit avoidance is to use artificial neural networks. In Reference [38], polynomial neural network based architecture is used for the main rotor blade stall limit cuing using equivalent retreating indicated tip speed (ERITS) as the limit parameter. With dynamic trim method proposed in Reference [30], an artificial neural network models the mapping between the controls and the limit parameter in order to estimate the quasi-steady state value of a limit parameter. Dynamics trim method, however, requires accurate models of dynamic trim characteristics, which means that large amount of dynamic trim data is required for the adequate training of neural networks.

Alternatively, an adaptive dynamics trim method was proposed in Reference [106], and is used. Instead of generating dynamics trim data to train a neural network off-line, an approximate first order model of the limit parameter dynamics is used in conjunction with an adaptive single hidden layer neural network. The dynamic trim method, however, relies on only steady-state-response limits, not transient-response limits. For example, steady-state-response limits are those limit parameters that reach their maximum values in steady state (load factor, angle of attack, etc.), while transient-response limits are those limit parameters that reach their maximum values in the

transient phase (flapping angle, hub moment, etc.). Envelope protection with peak-response estimate method [29] and nonlinear response function method [83] are proposed for those transient-response limit parameters. In addition, reactive envelope protection was proposed for autonomous envelope protection system [97][98]. Like the reactive obstacle avoidance method, this method considers the limit boundary as an obstacle, and generates a safe response profile of the limit parameter. Since it does not depend on the type of limit parameters, the reactive method is valid for both steady-state-response limits and transient-response limits.

1.5 Thesis Objectives and Outline

The goal of this thesis is to design mission-based guidance systems for autonomous UAVs, to integrate them with the other control subsystems (which are assumed to be available), and to evaluate the entire system in realistic simulations and/or in flight tests.

Chapter 2 presents mathematical preliminaries used in this thesis. Definitions and theorems from neural network (NN) approximation theory and numerical methods to solve optimal control problems are presented. The idea is to summarize basic mathematical results that are important to the development and presentation of the main ideas of the thesis.

Chapter 3 presents an adaptive approach to autonomous formation flight (AFF) of UAVs. It is assumed that the formation consists of a leader and a follower in a two-dimensional horizontal plane. The leader determines its own maneuver, and the follower is expected to maintain relative position with respect to the maneuvering leader. This approach assumes that the true values of range and the line-of-sight (LOS) angle are available for feedback. NN based adaptation is included in the guidance law design to compensate for the unknown leader maneuvers in the LOS kinematics. Simulation results

using the Georgia Tech UAV simulation tool (GUST) are presented for different leader maneuvers. In addition, results from flight demonstration using a ground vehicle as a leader are presented to show the efficacy of the adaptive approach by comparing the performance with a proportional guidance approach.

Chapter 4 develops a method to integrate obstacle avoidance and envelope protection. A problem is formulated as minimum-time control problem with an external constraint (obstacle avoidance) and an internal constraint (envelope protection). The obstacle avoidance constraint is converted into a safe waypoint constraint along with an implicit requirement that the horizontal velocity should not have negative values. Once a time-optimal solution in real-time is obtained, the algorithm determines whether the avoidance maneuver should be initiated right away according to the obtained acceleration command solution by way of including a requirement that the vehicle must maintain its original flight path to the maximum extent possible. The ways to handle limit parameter constraints are different based on relative degrees of limit parameters. Simulation results using the GUST and the nonlinear trajectory generator (NTG) are presented for different obstacle sizes, different flight conditions, and different limit values.

Chapter 5 summarizes contributions of the thesis, conclusions drawn from the thesis, recommended future work related to the research in the thesis.

CHAPTER 2

MATHEMATICAL PRELIMINARIES

This chapter presents mathematical results and tools required in the development of guidance systems for autonomous formation flight and obstacle avoidance in the subsequent chapters.

2.1 Neural Networks as Universal Approximators

The term artificial neural networks (ANNs) have come to mean any architecture that has massively parallel interconnection of simple processors. From a systems theoretical point of view, a neural network can be considered as conveniently parameterized class of nonlinear map. During the 1980's and early 1990's, conclusive proofs were given by numerous authors that multilayer feedforward networks are capable of approximating any continuous function on a compact set in a very precise and satisfactory sense [16][31]. As a result, such networks found wide application in many fields, both for function approximation and pattern recognition.

In fact, neural networks are widely used in designing controls in order to handle some difficulties such as uncertainties. Uncertainties are one of the most difficult problems in developing control architecture for the complex systems. In adaptive control, it is assumed that the uncertainty is linearly/nonlinearly parameterized without knowing the bounds of uncertainty. Controller parameters are updated online using available system signals to approximate the uncertainty. Conventional adaptive control methods have shown most successful results in applications where the uncertainty is a linear parameterization of known basis functions. In most cases, however, a linear

parameterization of the uncertainty is not simply known. Neural network (NN) based adaptive control methods can overcome the drawback of conventional adaptive control approaches since NNs can approximate nonlinear maps to any desired degree of accuracy. In addition, the fact that various algorithms are currently available for adjustment of the parameters of the networks implies that they can deal with uncertainty, by realizing approximations of unknown mappings, from system input-output data.

Multilayer feedforward NN shown in Figure 3 was introduced in the 1980's for approximating continuous functions. Later, the radial basis function neural network (RBFN) shown in Figure 4 was proposed as a viable alternative. The *n* layer NN with input *u* and output *y* is described as

$$\sigma[\mathbf{W}_n \sigma[\mathbf{W}_{n-1} \cdots \sigma[\mathbf{W}_1 u] \cdots]] = y \tag{1}$$

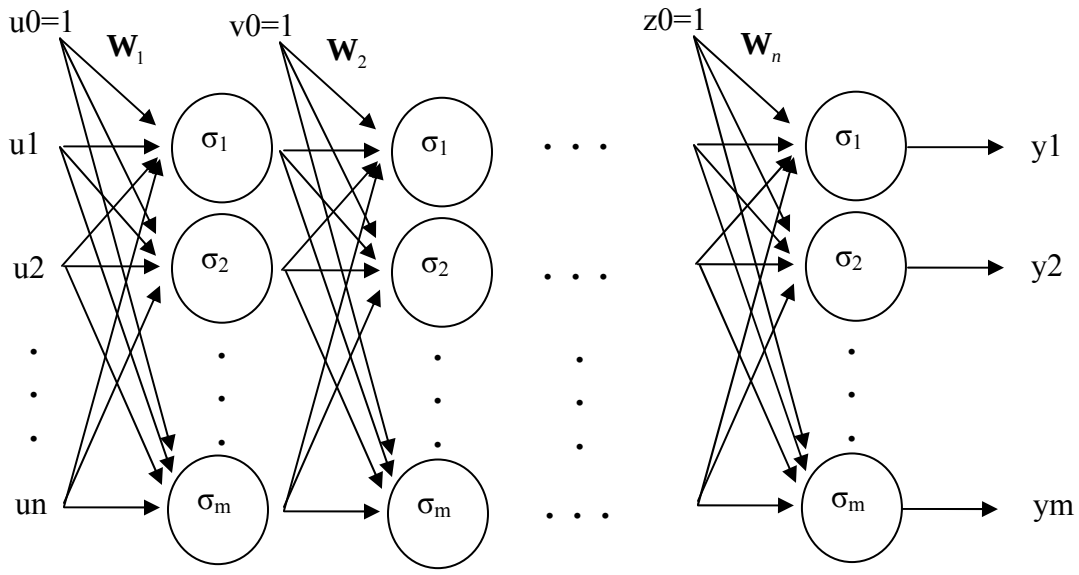


Figure 3. Multilayer feedforward neural network

where $\mathbf{W}_i$ is the weight matrix in the *i*th layer. $\sigma(\cdot)$ is a nonlinear operator with $\sigma(x) = [\sigma_1(x), \sigma_2(x), \cdots, \sigma_n(x)]^T$ where $\sigma_i(\cdot)$ is a smooth activation function, and

typically a sigmoid function is used for the activation function [16]. In Figure 4, the structure of the RBFN with input u and output y is shown. The output is described by the equation

$$y = \sum_{i=1}^{m} W_i R(u) + W_0 \tag{2}$$

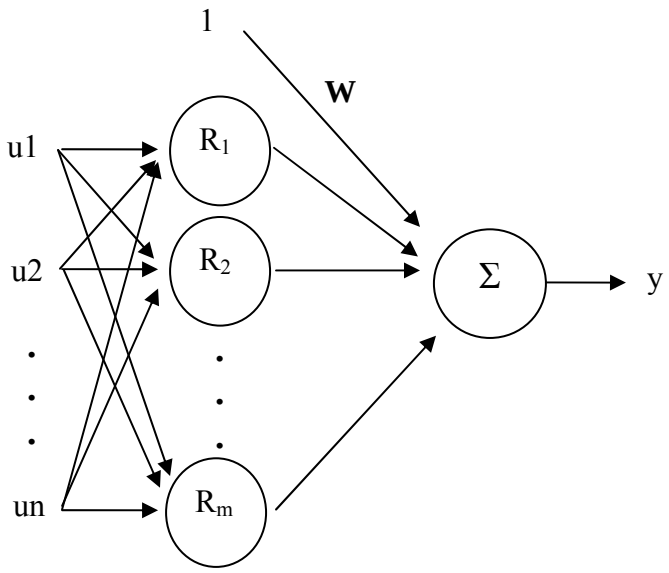


Figure 4. Radial basis function network

where W_i are the weight, and the function $R_i : \mathbf{R^n} \rightarrow \mathbf{R}$ has generally the form [58][90]

$$R_i(x) = \exp\left[-\sum_{j=1}^{n} \frac{\left(x_j - c_{ij}\right)^2}{2\varepsilon_{ij}} \right] \tag{3}$$

where $c_i = [c_{i1}, c_{i2}, \cdots, c_{in}]^T$ is the center of the *i*th receptive field and ε_{ij} is referred to as its width. The matrices $\mathbf{W}_i$ denoted the adjustable parameters of the multilayer feedforward NN, and the parameters W_i, the centers c_i and the width ε_{ij} constitute the adjustable parameters of RBFN. If, in some cases, c_i and ε_{ij} are chosen before the

network is used in a specific application, the mapping represented by RBFN becomes linear in the unknown parameters W_i.

While it has been shown that neural networks can approximate arbitrary continuous functions to any desired degree of accuracy, it is also true that polynomials, trigonometric series, splines, and orthogonal functions share the same properties. Hence, questions naturally arise as to why NNs should be preferred over such methods. Extensive computer studies carried out during the past years have revealed that NNs enjoy numerous practical advantages over conventional methods [69]. In view of their architecture, they are more fault tolerant and less sensitive to noise and they are more easily implementable in hardware because of the parameterization used.

Theorem 1[16][49]: Given arbitrary $\varepsilon^* > 0$, any continuous function $f(x): \mathbf{R^n} \to \mathbf{R^m}$ can be parameterized via a suitably chosen set of basis functions on a compact set $x \in \mathbf{D} \subset \mathbf{R^n}$

$$f(x) = \mathbf{W}^T \sigma(\mathbf{V}^T x) + \varepsilon(x), \quad \|\varepsilon(x)\| \le \varepsilon^* \tag{5}$$

where the basis function $\sigma(\cdot)$ is a shifted sigmoid function.

Theorem 2[33]: Assume that a *n* dimensional state vector $x(t)$ of an observable time-invariant systems

$$\begin{aligned} \dot{x} &= f(x) \\ y &= h(x) \end{aligned} \tag{6}$$

evolves on a *n* dimensional ball of radius r_0 in $\mathbf{R^n}$, $B_r = \{x \in \mathbf{R^n} \mid \|x\| \le r_0\}$. Also, assume that the system output *y* and its derivatives up to the order $(n-1)$ are bounded. Then

given arbitrary $\varepsilon^* > 0$, there exists a constant, bounded weights **W** and **V**, and a positive time delay $d > 0$, such that the function $f(x)$ can be approximated over the compact set B_r by nonlinearly parameterized NN

$$f(x) = \mathbf{W}^T \sigma(\mathbf{V}^T \bar{\mu}) + \varepsilon(\bar{\mu}), \|\varepsilon(\bar{\mu})\| \le \varepsilon^* \tag{7}$$

with bounded approximated derivatives $\|\bar{\mu}\| \le \mu^*$ and bounded parameters $\|\mathbf{W}\| \le W^*$ and $\|\mathbf{V}\| \le V^*$ using the input vector

$$\bar{\mu}(y(t), d) = \left[\Delta_d{}^0 y^T(t), \cdots, \Delta_d{}^{n-1} y^T(t)\right] \tag{8}$$

where $\Delta_d{}^0 y(t) = y(t), \Delta_d{}^k y(t) = \dfrac{\Delta_d{}^{k-1} y(t) - \Delta_d{}^{k-1} y(t-d)}{d}, k = 1, \cdots, n-1$ is uniformly bounded on B_r.

2.2 Optimal Control Problems under Consideration

Let $x \in \mathbf{R^n}$ denote the state of the system and $u \in \mathbf{R^m}$ the input of the system

$$\dot{x} = f(x, u) \tag{9}$$

where all vector fields and functions are real-analytic. It is desired to find a trajectory of that minimizes the performance index

$$J = \phi(x_f, u_f, t_f) + \int_{t_0}^{t_f} L(x, u, t) dt \tag{10}$$

subject to a vector of n_0 initial time constraints, n_f final time constraints, and n_p path constraints,

$$
\begin{aligned}
b_{0,l} &\le \psi_0(x_0,u_0) \le b_{0,u} \\
b_{f,l} &\le \psi_f(x_f,u_f) \le b_{f,u} \\
b_{p,l} &\le S(x,u,t) \le b_{p,u}
\end{aligned}
\tag{11}
$$

The functions ψ_0, ψ_f, S are assumed to be at least C^2 on appropriate dense open set. The final time t_f could be either fixed or free.

References [9][36][55] derive the necessary conditions using the calculus of variations. Pontryagin minimum principle shows that finding an extremal solution is equivalent to the requirement that the variational Hamiltonian be minimized, with respect to all admissible inputs.

Defining the Hamiltonian *H* and the auxiliary function Φ and Ξ

$$
\begin{aligned}
H(x,u,\lambda,\mu,t) &= L(x,u,t) + \lambda^T f(x,u) + \mu S^{(q)}(x,u,t) \\
\Xi(x_0,u_0,t_0,\nu_0) &= \nu_0{}^T \psi_0(x_0,u_0) \\
\Phi(x_f,u_f,t_f,\nu_f) &= \phi_f(x_f,u_f,t_f) + \nu_f{}^T \psi_f(x_f,u_f,t_f)
\end{aligned}
\tag{12}
$$

where the $\lambda : [t_0,t_f] \to \mathbf{R}^{\mathbf{n}}$, $\nu_0 \in \mathbf{R}^{\mathbf{n_0}}$, $\nu_f \in \mathbf{R}^{\mathbf{n}_f}$ and $\mu : [t_0,t_f] \to \mathbf{R}$ are Lagrange multipliers. The number of time derivative *q* of *S* such that control input *u* explicitly appears and $\frac{\partial S^{(q)}}{\partial u} \neq 0$ is denoted by $S^{(q)}(x,u,t)$. Assuming that $S(x,u,t) = 0$ on $[t_1,t_2]$, we also need the entry condition as follows:

$$
N(x(t_1),t_1) = \begin{bmatrix} S(x(t_1),t_1) \\ S^{(1)}(x(t_1),t_1) \\ \vdots \\ S^{(q-1)}(x(t_1),t_1) \end{bmatrix} = 0
\tag{13}
$$

Details of dealing with inequality constraints on the state space can be found in Reference [10] and [92]. The Lagrange multiplier $\pi \in \mathbf{R}^q$ will be associated with the constraints in equation 13.

Without loss of generality, it is assumed that there is one control input ($m = 1$). An optimal solution must satisfy the necessary conditions

$$\dot{x} = \frac{\partial H}{\partial \lambda} \tag{14}$$

$$\dot{\lambda} = -\frac{\partial H}{\partial x} \tag{15}$$

$$\lambda\left(t_1^{-}\right) = \lambda\left(t_1^{+}\right) + \pi^T \left.\frac{\partial N}{\partial x}\right|_{t=t1} \tag{16}$$

$$H\left(t_1^{-}\right) = H\left(t_1^{+}\right) - \pi^T \left.\frac{\partial N}{\partial t}\right|_{t=t1} \tag{17}$$

$$\lambda\left(t_f\right) = \left.\frac{\partial \Phi}{\partial x}\right|_{t=t_f} \tag{18}$$

and if the final time t_f is free the following condition, which is referred to as the transversality condition should be included.

$$\left.\left(\frac{\partial \Phi}{\partial t} + H\right)\right|_{t_f} = 0 \tag{19}$$

Unless the system equations, the performance index, and the constraints are quite simple, numerical methods may be needed to solve optimal programming and control problems. However, the amount of numerical computation required for even a relatively simple problem is forbidding if it must be done by hand. This is why the calculus of

variations found very little use in engineering and applied science until quite recently. The development of efficient and high-performance computers has drastically changed this situation. It is now possible to solve complicated optimal control problems in a reasonable length of time and at a reasonable cost. Since optimal control problems are at least two-point boundary value problems and, in some cases, multiple boundary value problems, finding solution to these problems is not a trivial extension of finding solutions to initial value problems. As shown before, the problem is to find the state variables, the Lagrange multiplier, and the control variables while simultaneously satisfying the system differential equations, the Euler-Lagrange differential equations, the optimality conditions, and the initial and final boundary conditions.

Various numerical methods to solve optimal control problems can be found in Reference [4]. In general, these methods can be categorized as direct methods or indirect methods. Indirect methods are based on the calculus of variations and the minimum principle. In the direct approach, the optimal control problem is transformed into a nonlinear programming problem (NLP). Direct methods are generally more robust to the initial solution guess than indirect methods. In addition, unlike indirect methods, direct methods do not require an a priori specification of the switching structure when inequality state constraints are present.

The NLP can be stated as the following:

$$\begin{aligned} &\min F(z) \\ &\text{subject to } b_{l,j} \le c_j(z) \le b_{u,j} \quad j = 1, \cdots, m \end{aligned} \tag{20}$$

where $b_{l,j}$ and $b_{u,j}$ represent lower and upper bounds of the constraint function c_j, respectively. Sequential Quadratic Programming (SQP) is one of the most popular methods to solve the NLP. At each iteration of an SQP method, one solves a quadratic program subproblem that models equation 20 locally at the current iteration. The solution to each iteration is used as a search direction to determine the next iteration.

Commercially available solvers for the NLP include NPSOL [24], SNOPT [25], CFSQP [54], etc.

The rest of this section discusses the conversion techniques from optimal control problems to the nonlinear programming problems. A reliable method to convert an optimal control problem to a nonlinear programming problem is collocation. The first step in the collocation method is to break time domain into smaller intervals

$$t_0 < t_1 < \cdots < t_N = t_f \tag{21}$$

Decision variables in the nonlinear programming then becomes the values of the state and the control at the grid point, namely,

$$z = \left(u_0, u_1, u_2, \cdots, u_N, x_1, x_2, \cdots, x_N\right) \tag{22}$$

The key notion of collocation method is to replace the original system with a set of discrete constraints $\zeta_i = 0$, which are imposed on each interval of discretization. If we assume there are also a state variable constraint $S\left(x\left(t\right)\right) = 0$, initial condition $\psi_0\left(x\left(t_0\right)\right) = 0$, and final constraint $\psi_f\left(x\left(t_f\right)\right) = 0$, then the complete set of nonlinear programming constraints are

$$c\left(z\right) = \begin{bmatrix} \psi_0\left(x\left(t_0\right)\right) \\ S\left(x_0\right) \\ \zeta_0 \\ S\left(x_1\right) \\ \vdots \\ S\left(x_{N-1}\right) \\ \zeta_{N-1} \\ S\left(x_N\right) \\ \psi_f\left(x\left(t_N\right)\right) \end{bmatrix} = 0 \tag{23}$$

It is apparent that the nonlinear programming problem that results from this formulation is very large, rendering a real-time implementation difficult.

Another way to convert optimal control problems to nonlinear programming problems is the adjoint method. The adjoint method is a direct method that uses a combination of nonlinear programming and shooting. In contrast to the collocation method, the adjoint method has significantly less number of decision variables. The trade-off is that integration has to be performed backward in time on an adjoint system for each constraint. Reference [9] states that numerical integration required to find the adjoint is quite stable since integration is carried out in backward time. Applying the adjoint lemma [43] and constructing a control input history such that the cost function is decreasing, the gradient of the cost function, the gradient of the end point constraint, and the gradient of the state inequality constraint can be obtained. Detailed expressions are shown in Reference [61].

CHAPTER 3

ADAPTIVE GUIDANCE DESIGN FOR AUTONOMOUS FORMATION FLIGHT

In this chapter, adaptive guidance logic for autonomous formation flight of UAVs is proposed and developed. Autonomous formation flight is a mechanism for achieving a pre-specified formation among a group of unmanned aerial vehicles. Using measurements of the line-of-sight range and angles, an adaptive guidance law is formulated for the follower that generates velocity commands so that the follower maintains a prescribed range from the leader in the presence of leader maneuvers. The method presented uses artificial neural networks (NN) as an adaptive element in the guidance controller. A NN has been used in order to compensate for the effect of uncertainties in controlled output dynamics. Software-in-the-loop simulation and flight test evaluations of an adaptive velocity-command guidance controller for autonomous formation flight are presented.

3.1 Introduction

Formation control of multiple unmanned aerial vehicles (UAVs) has attracted significant attention from the UAV research community. The objective of formation control is to obtain a group of autonomous agents to move together in a desired formation and accomplish desired tasks such as reconnaissance, surveillance, precision strike, etc. In most autonomous formation flight (AFF) designs, communications between UAVs in formation is required. In general, information about a vehicle in formation is broadcast to the entire group or only to the adjacent vehicles in close proximity. Recently, many

different approaches to communication for formation control are introduced. If active communication is available, formation flight controller shows desirable performance [56][100]. Reference [100] shows that wireless communication network can be used in order to get position and velocity data of each vehicle. Globally stable AFF derived in Reference [56] uses position, velocity, heading, and leader's input data. A closed form guidance method for formation flight and its stability analysis are presented in Reference [89]. This guidance law is only valid under the assumption that all information about the speed and the path angle of the nearest vehicle is available. Even though an active communication link between the vehicles can increase the performance of the AFF control, it requires the receiver and transmitter combination to be healthy for mission completion. Since a failure of any part of the communication equipment results in a failure of entire task, passive detection methods would be much preferred.

One of the methods for passive detection is to use the wake of the leading aircraft, which can characterize the relative position from the leader. In Reference [73], a neural network (NN) is used to find the relationship between wake and position from the leader. Another passive method for detecting nearby vehicles is to use a vision sensor [3][26]. An omni directional camera in conjunction with image processing algorithms and nonlinear filters, such as an Extended Kalman Filter (EKF), can be used to estimate relative position, velocity and attitude with respect to a nearby vehicle. References [42] and [85] present an approach for formation flight controller design for fixed-wing aircraft using information on relative positions obtained via image processing. It is assumed that every vehicle in formation knows its own speed and heading and can measure line-of-sight (LOS) range and angle to an adjacent aircraft in formation. Formation guidance commands are given in the form of velocities [42] or accelerations [85]. In Reference [95], a guidance law using only LOS angle and angular rate is proposed. The developed formation control scheme divides the formation flight procedure into an approach mode

and a guidance mode. However, no explanation is given in Reference [95] on how LOS rate measurements can be derived via passive methods.

Assuming that the only information available is measurements of LOS range and angle to a leader vehicle and own-aircraft navigation data, there is a need to have a mechanism to compensate for the unmodeled leader aircraft maneuvers. One way to do this is to incorporate models of all possible leader maneuvers, and another way is to directly compensate the effect of those maneuvers on the guidance law via an adaptive mechanism. Neural net based adaptive architectures were developed in conjunction with an acceleration command guidance [67] and a velocity command guidance [68] for formation flight involving a rotary wing UAV as the follower vehicle. Since the design of the guidance system is independent of the autopilot controller, both the acceleration command guidance system and the velocity command guidance system will be developed and compared using an integration of the GTMax autopilot controller. One may also consider the position command guidance system.

3.2 Adaptive Guidance for Formation Flight

3.2.1 Velocity command guidance system

Figure 5 shows the basic geometry associated with a 2-D formation flight, which consists of a leader (L) and a follower (F). The terms, V_L and ψ_L, represent the speed and the heading angle of the leader, respectively. Likewise, V_F and ψ_F are the follower's speed and heading angle, respectively. The vehicle kinematics is modeled as follows:

$$\begin{aligned} \dot{x} &= V\cos\psi \\ \dot{y} &= V\sin\psi \end{aligned} \tag{24}$$

The measured joint performance variables are the relative distance R and the LOS angle λ_A which are defined as follows:

$$R = \sqrt{(x_L - x_F)^2 + (y_L - y_F)^2}$$
$$\lambda_A = \tan^{-1}\left(\frac{y_L - y_F}{x_L - x_F}\right) \tag{25}$$

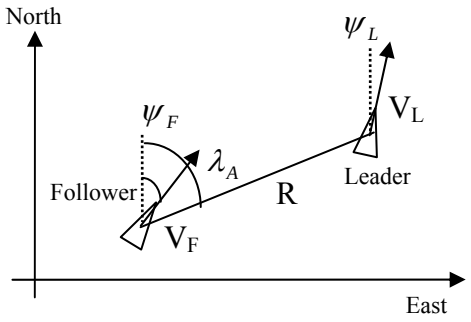


Figure 5. Configuration of formation flight in the horizontal plane.

where *x* and *y* represent the position along the North and the East axes, respectively. The follower also knows its own position and velocity. The main objective of the AFF control system is to maintain relative distance as commanded.

Based on the fact that the control law design for the formation flight problem exhibits a two time-scale feature like other conventional flight trajectory control problems [67][85], the overall architecture of the AFF control system consists of a guidance system and an autopilot system. Reference [88] shows experimental results using this concept of controller design. The alternative method uses an integrated approach wherein the guidance loop and the autopilot controller are designed simultaneously [46][91]. In most actual flight applications, the separate inner and outer loop design approach is commonly taken because it is usually simpler and well-designed autopilot controllers are available.

It is assumed that inputs to the guidance system are the commanded relative positions and outputs are the velocity commands. A leader is chosen to direct the

formation and can be either a piloted vehicle or another UAV. Thus the objective of the AFF controller will be to maintain a prescribed relative range with respect to the leader in the presence of leader maneuvers. The only information about the leader is the LOS data assumed to be available by means of a passive sensor system. In order to determine the velocity commands from the measured joint performance variables, we will mainly use the theory for multi-input multi-output (MIMO) adaptive output feedback, which is detailed in References [32] and [33]. Even though this thesis focuses on formation flight in the horizontal plane, the design of the adaptive guidance system for formation flight in the three-dimensional space are developed in Reference [84].

Relative distance control

The deviation angle is defined by

$$\theta = \psi_F - \lambda_A \tag{26}$$

Equation 26 represents difference between the LOS angle and the heading angle of the follower. Since a pure following situation means that the deviation angle becomes zero, the heading command of the follower is set equal to the LOS angle as follows:

$$\psi_{F,com} = \lambda_A \tag{27}$$

Starting with equation 27, the magnitude of velocity vector will be determined. The time derivative of the relative distance is given by

$$\dot{R} = \frac{1}{R}\left\{\left(x_L - x_F\right)\left(\dot{x}_L - \dot{x}_F\right) + \left(y_L - y_F\right)\left(\dot{y}_L - \dot{y}_F\right)\right\} \tag{28}$$

Using the definition of the LOS angle, we can rewrite equation 28 as follows:

$$\dot{R} = \cos(\lambda_A)\left(\dot{x}_L - \dot{x}_F\right) + \sin(\lambda_A)\left(\dot{y}_L - \dot{y}_F\right) \tag{29}$$

From equation 24, equation 29 can be written as

$$\dot{R} = -V_F \cos(\psi_F - \lambda_A) + V_L \cos(\psi_L - \lambda_A) \tag{30}$$

The first term in the right-hand side in equation 30 is the follower velocity along the LOS and the second term is the leader velocity along the LOS, which is unknown to the follower. Furthermore, the direction of desired follower velocity is determined as equation 27, and the only thing left to be determined is its magnitude. By defining tracking error as $e = R - R_c$, we can obtain the error dynamics as follows:

$$\begin{aligned} \dot{e} &= \dot{R} - \dot{R}_c \\ &= -V_F \cos(\psi_F - \lambda_A) + V_L \cos(\psi_L - \lambda_A) - \dot{R}_c \end{aligned} \tag{31}$$

where R_c is the filtered range command. Hence, the speed command for the follower can be obtained as

$$V_{F,com} = \frac{-\dot{R}_c + \alpha e + \nu_{ad}}{\cos(\psi_F - \lambda_A)} \tag{32}$$

where $\alpha > 0$ is a design parameter, and ν_{ad} is the output of an adaptive neural network (NN), designed to compensate for the leader velocity along the LOS.

Note: Equation 32 raises the question on how the speed command is computed in case the denominator goes to zero. This happens when the follower heading is perpendicular to the LOS and this implies range cannot be regulated anymore using speed control. One easy way to resolve this potential issue is to set $V_{F,com} = 0$ when $|\psi_F - \lambda_A| \approx \pi/2$, and only use heading control, i.e., $\psi_{F,com} = \lambda_A$.

If the speed command for the follower in equation 32 is achieved, then the error dynamics in equation 31 reduces to

$$\dot{e} = -\alpha e - \nu_{ad} + V_L \cos(\psi_L - \lambda_A) \tag{33}$$

As shown in equation 33, the tracking error will converge to zero as long as the NN output effectively cancels out the leader velocity along the LOS.

Adaptive neural network design

A single-hidden-layer (SHL) NN is used to approximate the $V_L \cos(\psi_L - \lambda_A)$ term in equation 33. Reference [32] established a universal approximation for an unknown continuous function of states and control in a bounded, observable process using a memory unit of sampled input/output pairs. Figure 6 shows the generic structure of a SHL NN.

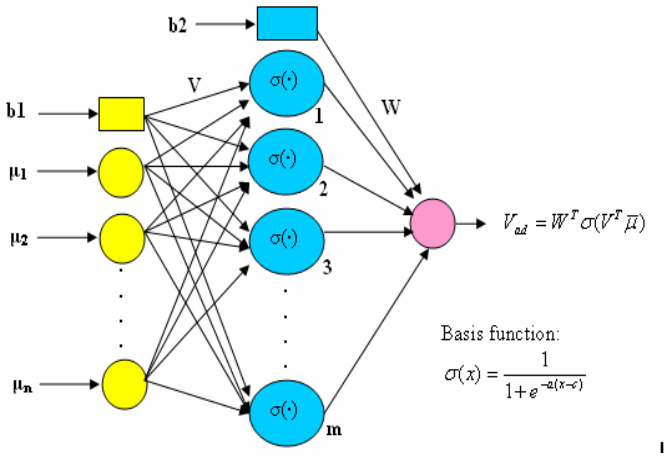


Figure 6. Generic structure of a SHL NN.

The SHL NN weights are updated on-line using the adaptive law given by [32], which is referred to as a sigma-modification rule.

$$\begin{aligned}\dot{\hat{W}} &= -\Gamma_W\left[2\left(\sigma-\sigma'\hat{V}\bar{\mu}\right)e^{T}PB+k_W\hat{W}\right]\\ \dot{\hat{V}} &= -\Gamma_V\left[2\bar{\mu}PBW^{T}\sigma'+k_V\hat{V}\right]\end{aligned} \tag{34}$$

where $\Gamma_W > 0$ and $\Gamma_V > 0$ are adaptation gains for the output layer and the hidden layer, respectively. In addition, σ' denotes the Jacobian matrix, P is the solution of the Lyapunov equation

$$\alpha P + P\alpha = Q \tag{35}$$

for some $Q > 0$, and $k_W > 0$ and $k_V > 0$ are the sigma modification gains. $\bar{\mu}$ represents the NN input vector, defined as

$$\bar{\mu} = \begin{bmatrix}1 & u_d^T & y_d^T\end{bmatrix}^T \tag{36}$$

where u_d and y_d are delayed input and delayed output vector, respectively, which are expressed as follows:

$$u_d(t) = \left[V_{F,com}(t)\ V_{F,com}(t-d)\ V_{F,com}(t-2d)\ \psi_{F,com}(t)\ \psi_{F,com}(t-d)\ \psi_{F,com}(t-2d)\right]^T \tag{37}$$

$$\begin{aligned}y_d(t) = [&R(t)\ R(t-d)\ R(t-2d)\ \lambda_A(t)\ \lambda_A(t-d)\ \lambda_A(t-2d)\\ &V_F(t)\ V_F(t-d)\ V_F(t-2d)\ \psi_F(t)\ \psi_F(t-d)\ \psi_F(t-2d)]^T\end{aligned} \tag{38}$$

Pseudo-control hedging (PCH) [39] is introduced to protect the adaptive law from effect due to actuator rate and position limits, and unmodeled actuator dynamics. The main idea behind the PCH is to modify the reference command in order to prevent the adaptive element from adapting to these actuator characteristics. The reference model is hedged by an amount equal to the difference between the commanded and the achieved pseudo-control. Note that since the velocity command is used, the velocity autopilot dynamics appear like actuator dynamics to the adaptive guidance law. The hedge signal is

a deficit in the velocity achieved along the LOS with respect to the commanded LOS velocity and is given by

$$\nu_h = -V_{F,com}\cos\left(\psi_{F,com} - \lambda_A\right) - \left[-V_F\cos\left(\psi_F - \lambda_A\right)\right] \tag{39}$$

Since the relative degree of the range with respect to the velocity is 1, a first-order reference model for the filtered range command is used as shown below:

$$\begin{aligned} \nu_{crm} &= \tau\left(R_{com} - R_c\right), \quad \tau > 0 \\ \dot{R}_c &= \nu_{crm} - \nu_h \end{aligned} \tag{40}$$

where R_{com} is the commanded range, R_c is the reference range obtained by filtering R_{com} through the first-order, hedged reference model in equation 40.

3.2.2 Acceleration command guidance system

Now, this section develops an acceleration command guidance system. The vehicle acceleration equations are augmented as follows:

$$\begin{aligned} \dot{\psi} &= \frac{a_1}{V} \\ \dot{V} &= a_2 \end{aligned} \tag{41}$$

It is assumed that inputs to the guidance system are the commanded relative positions and outputs are the acceleration commands. Like in Section 3.2.1, the objective of the AFF controller will be to maintain a prescribed relative range with respect to the leader in the presence of leader maneuvers. The only information about the leader is the LOS data assumed to be available by means of a passive sensor system. In order to determine the acceleration commands from the measured joint performance variables, the idea of multi-input multi-output (MIMO) adaptive output feedback is also used.

Deviation angle control

Starting From equation 26, the time rate of the deviation angle is given by

$$\dot{\theta} = \dot{\psi}_F - \dot{\lambda}_A \tag{42}$$

From equations 25 and 41, equation 42 can be rewritten as

$$\dot{\theta} = \frac{a_{1F}}{V_F} - \frac{1}{R}\left\{-V_F \sin\left(\psi_F - \lambda_A\right) + V_L \sin\left(\psi_L - \lambda_A\right)\right\} \tag{43}$$

It can be noticed from equation 43 that relative distance of the deviation angle is one. By defining tracking error $e_\theta = \theta - \theta_c$, we can obtain the error dynamics as follows:

$$\dot{e}_\theta = \frac{a_{1F}}{V_F} - \frac{1}{R}\left\{-V_F \sin\left(\psi_F - \lambda_A\right) + V_L \sin\left(\psi_L - \lambda_A\right)\right\} - \dot{\theta}_c \tag{44}$$

where θ_c is the filtered deviation angle command. Hence, the acceleration command $a_{1F,com}$ for the follower can be obtained as

$$a_{1F,com} = V_F\left(\dot{\theta}_c - \alpha\, e_\theta - \nu_{1,ad}\right) \tag{45}$$

where $\alpha > 0$ is a design parameter, and $\nu_{1,ad}$ is the output of neural networks associated with the LOS angle dynamics. If the acceleration command given by equation 45 is achieved, the error dynamics in equation 44 reduces to

$$\dot{e}_\theta = -\alpha\, e_\theta - \nu_{1,ad} - \frac{1}{R}\left\{-V_F \sin\left(\psi_F - \lambda_A\right) + V_L \sin\left(\psi_L - \lambda_A\right)\right\} \tag{46}$$

As shown in equation 46, the tracking error will converge zero as long as the NN output effectively cancels out the last term in the error dynamics.

Relative distance control

From equation 30, the second time derivative of the relative distance is given by

$$\begin{aligned}\ddot{R} = &-\dot{V}_F \cos(\psi_F - \lambda_A) + V_F(\dot{\psi}_F - \dot{\lambda}_A)\sin(\psi_F - \lambda_A) \\ &+ \dot{V}_L \cos(\psi_L - \lambda_A) - V_L(\dot{\psi}_L - \dot{\lambda}_A)\sin(\psi_L - \lambda_A)\end{aligned} \tag{47}$$

From equation 41, equation 47 can be rewritten as

$$\begin{aligned}\ddot{R} = &-a_{2F} \cos(\psi_F - \lambda_A) + a_{1F} \sin(\psi_F - \lambda_A) + a_{2L} \cos(\psi_L - \lambda_A) - a_{1L} \sin(\psi_L - \lambda_A) \\ &+ \frac{1}{R}\{V_L \sin(\psi_L - \lambda_A) - V_F \sin(\psi_F - \lambda_A)\}^2\end{aligned} \tag{48}$$

From equation 48, relative degree of the relative distance is two. The acceleration command $a_{2F,com}$ for the follower can be obtained as

$$a_{2F,com} = \frac{-1}{\cos(\psi_F - \lambda_A)}\left\{-a_{1F,com} \sin(\psi_F - \lambda_A) - \nu_{dc} - \nu_{2,ad} + \ddot{R}_c\right\} \tag{49}$$

where ν_{dc} is the output of the linear compensator, and $\nu_{2,ad}$ is the output of the NN associated with the relative distance dynamics. If equation 49 is achieved, then the error dynamics $e_R = R - R_c$ becomes

$$\begin{aligned}\ddot{e}_R = &-\nu_{dc} - \nu_{2,ad} + a_{2L} \cos(\psi_L - \lambda_A) - a_{1L} \sin(\psi_L - \lambda_A) \\ &+ \frac{1}{R}\{V_L \sin(\psi_L - \lambda_A) - V_F \sin(\psi_F - \lambda_A)\}^2\end{aligned} \tag{50}$$

Since the relative degree in this case is two, the following linear compensator is introduced to stabilize the dynamics in equation 50

$$\begin{aligned}\dot{\eta} &= a\eta + be_R \\ \nu_{dc} &= c\eta + de_R\end{aligned} \tag{51}$$

where $a, b, c,$ and d are design parameters.

Adaptive neural network design

Like in Section 3.2.1, a SHL NN is used to approximate unknown terms in both the deviation angle dynamics and the relative distance dynamics. The sigmoid-adaptation rule given by equation 34 is used. Here, a matrix $P>0$ is a solution to the Lyapunov equation

$$\overline{A}^T P + P\overline{A} + Q = 0 \tag{52}$$

where any $Q>0$, and $\overline{A}$ is given by

$$\overline{A} = \begin{bmatrix} A - BdC & -Bc \\ bC & a \end{bmatrix}, A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{53}$$

Note that $a, b, c,$ and d in equation 51 should be selected such that $\overline{A}$ is Hurwitz. A detailed stability analysis can be found in Reference [32].

Also, the NN input vector is defined as the same in equation 36, and delayed input vector and delayed output vector are expressed as follows:

$$\begin{aligned} u_d(t) = [&a_{1F,com}(t)\ a_{1F,com}(t-d)\ a_{1F,com}(t-2d)\ a_{1F,com}(t-3d) \\ &a_{2F,com}(t)\ a_{2F,com}(t-d)\ a_{2F,com}(t-2d)\ a_{2F,com}(t-3d)]^T \end{aligned} \tag{54}$$

$$\begin{aligned} y_d(t) = [&R(t)\ R(t-d)\ R(t-2d)R(t-3d)\lambda_A(t)\ \lambda_A(t-d)\ \lambda_A(t-2d)\lambda_A(t-3d) \\ &V_F(t)\ V_F(t-d)\ V_F(t-2d)V_F(t-3d)\ \psi_F(t)\ \psi_F(t-d)\ \psi_F(t-2d)\psi_F(t-3d)]^T \end{aligned} \tag{55}$$

3.3 Integrated Simulation using the GTMax

The GTMax rotary wing test bed (see Figure 7) is a modified Yamaha RMax helicopter that uses a unique integrated simulation and flight test architecture. Detailed description of the GTMax hardware configuration can be found in Reference [41].

Figure 7. Georgia Tech helicopter UAV: GTMax.

A block diagram representation of the GTMax trajectory following controller architecture [40] is shown in Figure 8. The integration of the velocity-command guidance system with the GTMax controller is also shown in Figure 8. The first component in Figure 8 is the guidance loop for formation flight, which provides velocity commands to the flight controller as a function of time. The default flight controller is an adaptive NN trajectory following controller with 18 NN inputs, 5 hidden layer neurons, and 7 outputs for each of the 7 degrees of freedom. The 7 degrees of freedom include the usual 6 rigid-body degrees of freedom plus a degree of freedom for rotor RPM. This flight controller can also be configured as a conventional inverting controller. The navigation system is a 17 state Extended Kalman Filter that fuses information from 5 related sensors (GPS, IMU, sonar, radar, and magnetometer) to provide estimates of vehicle position, velocity, attitude, and terrain height. The flight controller determines actuator commands based on

the outputs of the navigation system and guidance commands. The GTMax helicopter model, the helicopter interface model, and sensor models have been developed as a simulation tool. This simulation tool is written primarily in C/C++ and has been developed to allow the test architecture to run on a high-end personal computer.

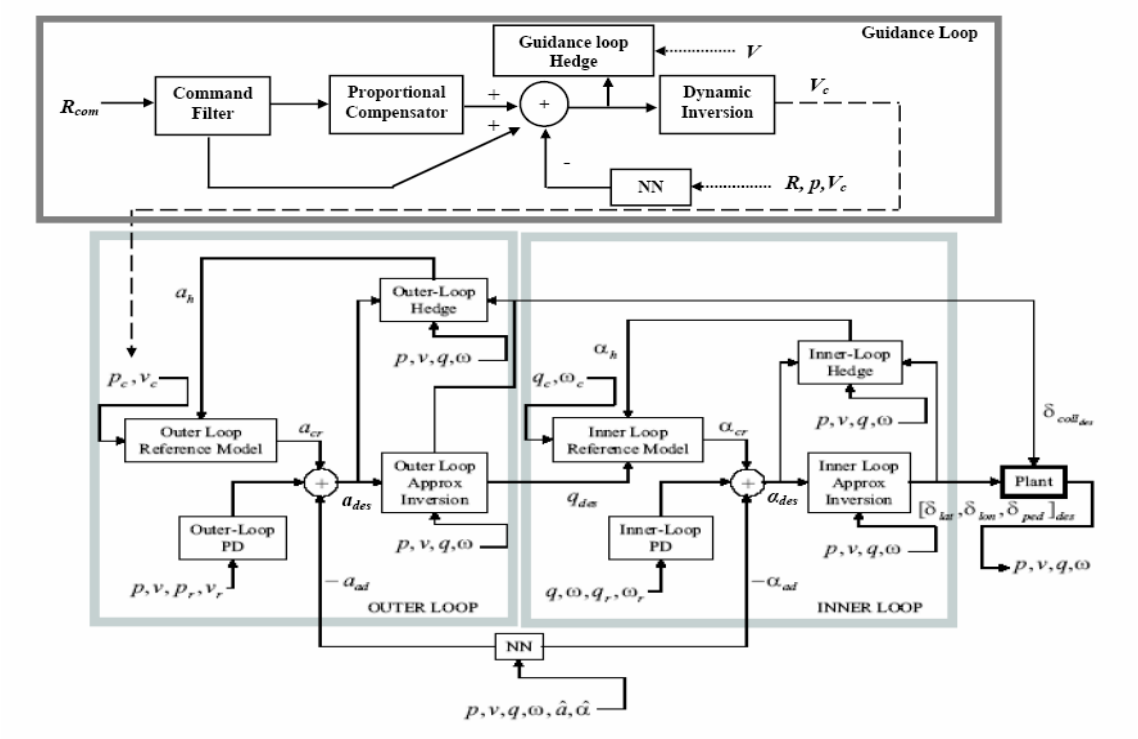


Figure 8. Integration of the adaptive velocity-command guidance architecture with GTMax controller.

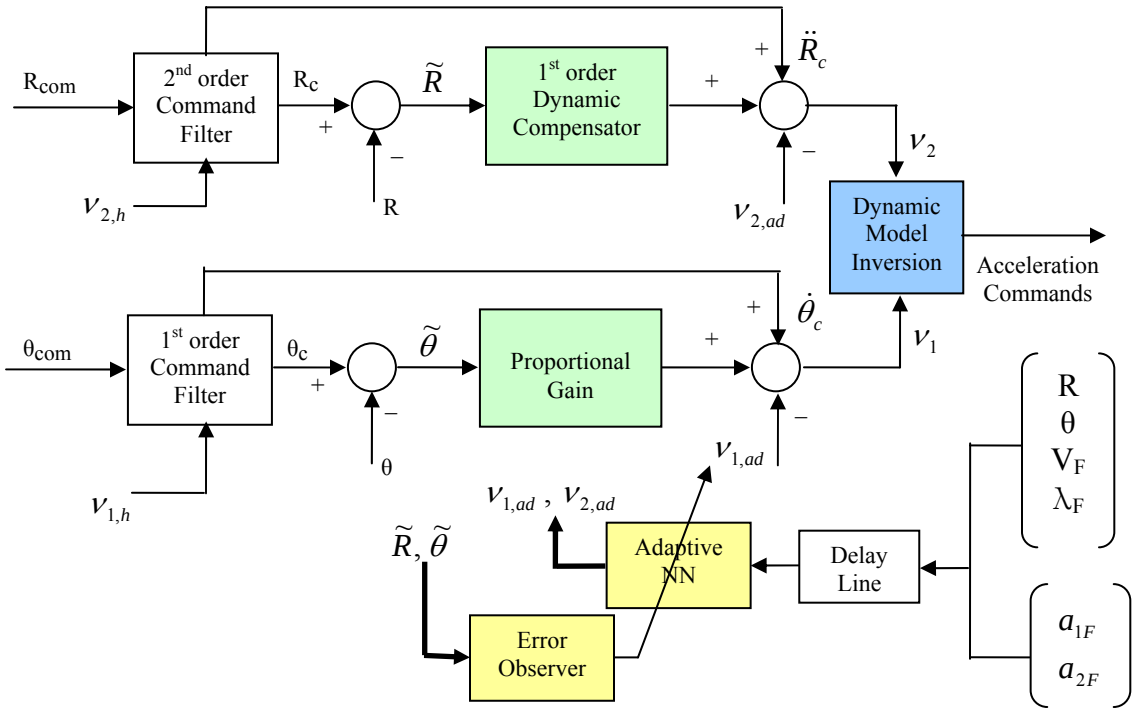


Figure 9. Block diagram of the adaptive acceleration-command guidance architecture.

The software-in-the-loop (SITL) simulation configuration refers to the combined simulation of the ground control station (GCS), onboard routines, and simulated sensors and vehicle dynamics on any desktop computer. In this configuration, all hardware is simulated to the level of its digital communication with other components. This configuration is useful for rigorous software testing without requiring any actual flight hardware. In the SITL architecture, shown in Figure 10, a communication link is established between the primary and secondary computers using routines contained within the data communications software.

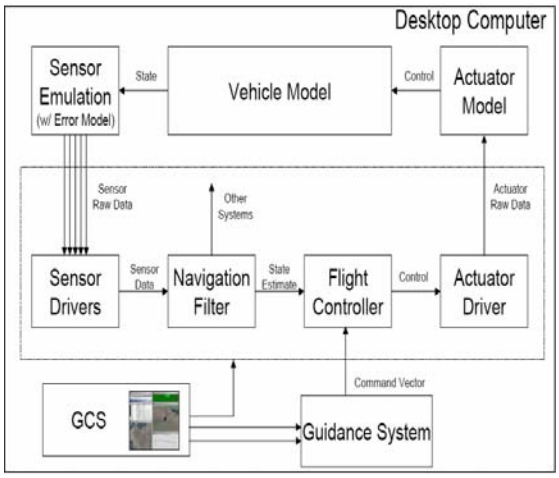


Figure 10. Software-in-the-loop simulation evaluation architecture.

3.3.1 Guidance Controller Parameters

Both software-in-the-loop (SITL) simulations and flight test evaluations make use of the same set of controller parameters. Tables 1 and 2 list the values of some of the important controller parameters for the adaptive velocity-command guidance loop and for the adaptive acceleration-command guidance loop, respectively. Table 3 lists controller bandwidths for the adaptive trajectory controller.

Table 1. Adaptive velocity-command guidance loop parameters used for SITL and flight test evaluations.

Proportional gain	1.0
Adaptation rate (inner layer)	0.1
Adaptation rate (outer layer)	0.1
No. of neurons (hidden layer)	5
No. of neurons (input layer)	25
Update rate	50 Hz

Table 2. Adaptive acceleration-command guidance loop parameters used for SITL evaluations.

Proportional gain	3.0
Adaptation rate (inner layer)	0.1
Adaptation rate (outer layer)	0.1
No. of neurons (hidden layer)	5
No. of neurons (input layer)	33
Eigenvalues of $\overline{A}$	-1.56 -0.72±2.22i
Update rate	50 Hz

Table 3. Trajectory controller bandwidth used in SITL and flight test evaluations.

x-, y-, z- loop bandwidths	2, 2.5, 3 rad/sec
Roll, pitch, yaw loop bandwidths	2.5, 2, 3 rad/sec
Update rate	50 Hz

3.3.2 Software-in-the-loop (SITL) Simulation Evaluation

Box-shaped maneuver

First, a box-shaped maneuver by the lead vehicle is used for the SITL evaluations. In this maneuver, the leader starts from rest and accelerates up to 30 *ft/s*. Then it makes three successive turns and returns to rest. Figure 11 shows a box-shaped trajectory performed by the lead vehicle. A UAV is commanded to maintain the relative distance equal to 100 *ft*. Since the UAV is initially located at the origin (0, 0), the initial range is 200 *ft*.

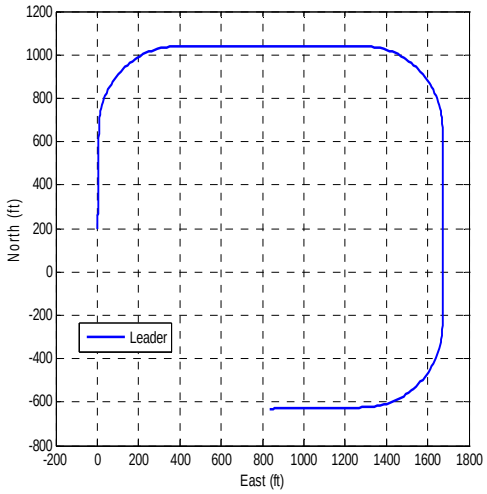


Figure 11. Formation box-shaped trajectory.

Figures 12, 13, and 14 show trajectory responses for the cases of velocity-command guidance without NN, velocity-command guidance with NN, and acceleration-command guidance with NN, respectively. It can be noticed from Figure 12 that there is an offset error in range tracking. In contrast, with NN switched on, the range response shows a good tracking performance. With adaptive acceleration command guidance system, it is seen from Figure 14 that there is the noticeable tracking error during the turning maneuver. This result is not surprising since acceleration commands from the guidance system are integrated twice to generate the needed position commands for the GTMax

trajectory controller. In contrast, in the velocity command guidance, the velocity command from the guidance system needs to be integrated only once to get the required position command. Therefore, the rest of this chapter focuses on the evaluation of the adaptive velocity command guidance system. Figure 15 explains the reason why the NN-based adaptive guidance law works better that the proportional error based linear guidance law. The output of the NN effectively cancels out the leader velocity along the LOS. In Figure 16, the inversion errors in the LOS angle dynamics and in the range dynamics are compared with the outputs of the NN.

Figures 17 and 18 show range tracking responses for the cases of $R_{com} = 50\ ft$ and $R_{com} = 25\ ft$, respectively. It can be noticed that with a smaller range command, the range tracking error increases during the turning maneuvers. This is caused by the fact that the effect of the leader heading change increases when the follower is flying much closer to the lead vehicle. In addition, the integration of the velocity command guidance system with the GTMax autopilot controller is investigated regarding the range tracking performance. Since velocity commands are determined at the guidance system, we need to integrate these commands to obtain position commands. By setting position feedback gains to zero, we try to represent the velocity tracking autopilot system without redesigning the GTMax autopilot system. Figure 19 shows the range tracking response for the case of the GTMax autopilot system without position feedback. It is seen that there is an offset error in range tracking for the case of a modified GTMax autopilot controller. Thus, by comparing Figures 13 and 19, it is inferred that velocity guidance system combined with the original GTMax autopilot system shows much preferable range tracking performance, compared to velocity guidance system combined with the GTMax autopilot system without position feedback.

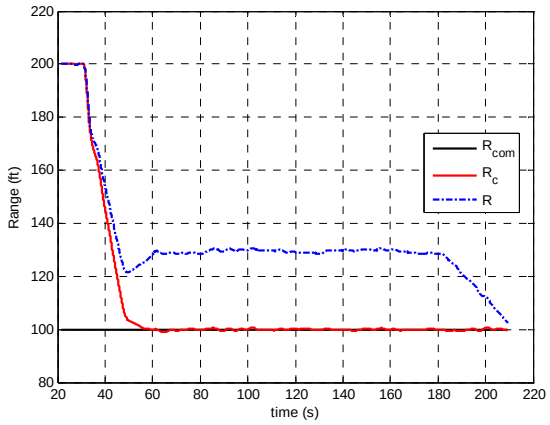


Figure 12. Range tracking response – velocity command guidance w/o NN

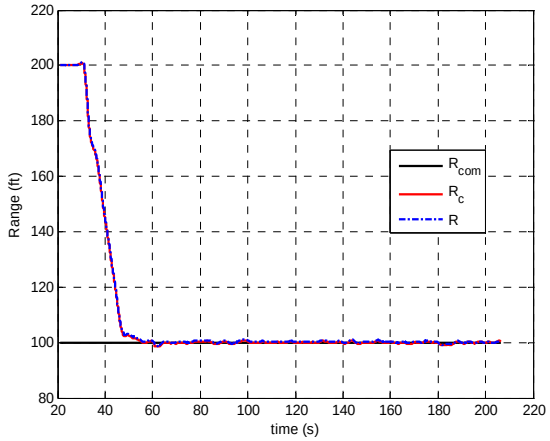


Figure 13. Range tracking response – velocity command guidance with NN

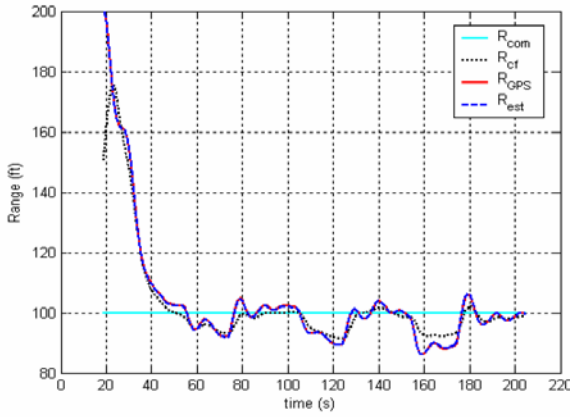


Figure 14. Range tracking response – acceleration command guidance with NN

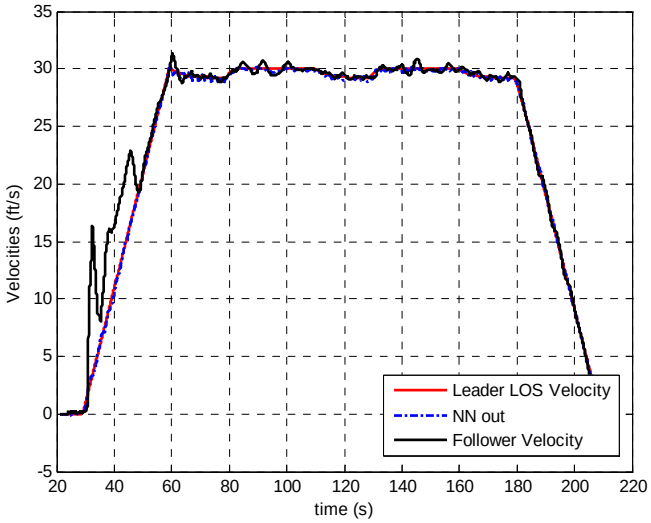


Figure 15. Leader LOS velocity vs. NN output – adaptive velocity command guidance

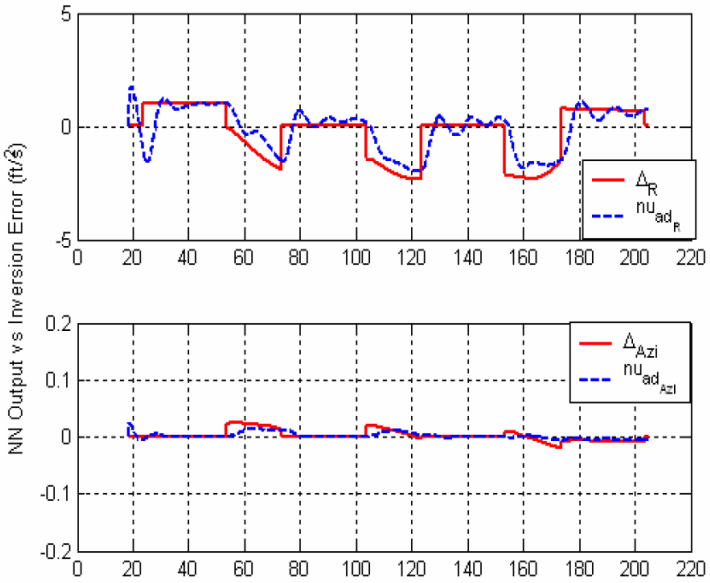


Figure 16. Inversion error vs. NN output – adaptive acceleration command guidance

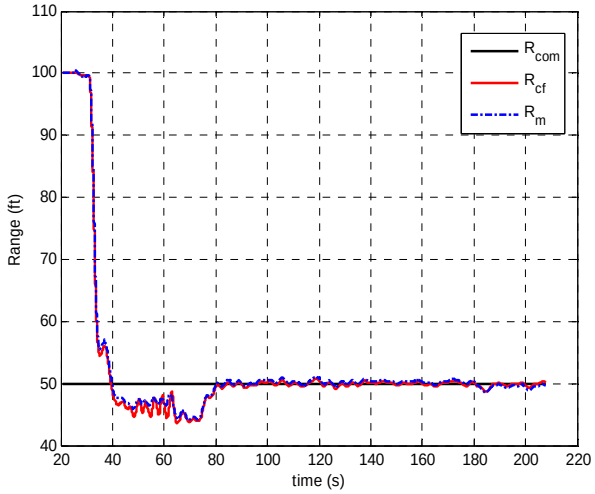


Figure 17. Range tracking response – R_{com} = 50 *ft*

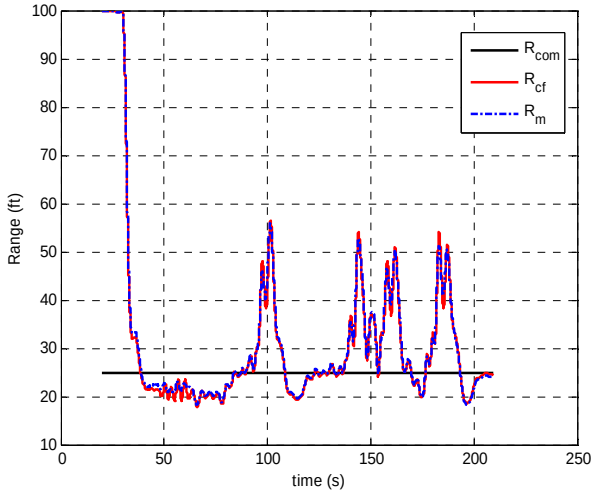


Figure 18. Range tracking response – R_{com} = 25 *ft*

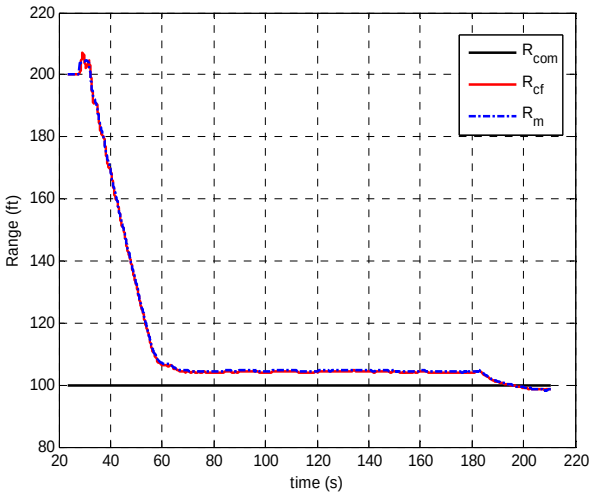


Figure 19. Range tracking response with the GTMax autopilot w/o position feedback

Racetrack maneuver and sudden stop

Next, a racetrack maneuver by the lead vehicle is used for the SITL simulation evaluations. In this maneuver, the lead vehicle is moving along an oval track at 30 *ft/s* speed, and the GTMax is commanded to maintain a relative distance equal to 100 *ft*. Figure 20 shows the trajectories of the leader and the follower, while Figure 21 shows range tracking performance with and without the NN adaptation. In Figure 21, R_{com} is range command, R_{cf} is hedged filtered range command and R is actual range response. During leader heading change or speed change maneuvers, there are small overshoots in the range response. With the adaptive component of the guidance law switched off, a steady-state error in the range tracking performance appears whose magnitude is proportional to the speed of the leader. With the adaptive NN switched on, the tracking error is significantly reduced.

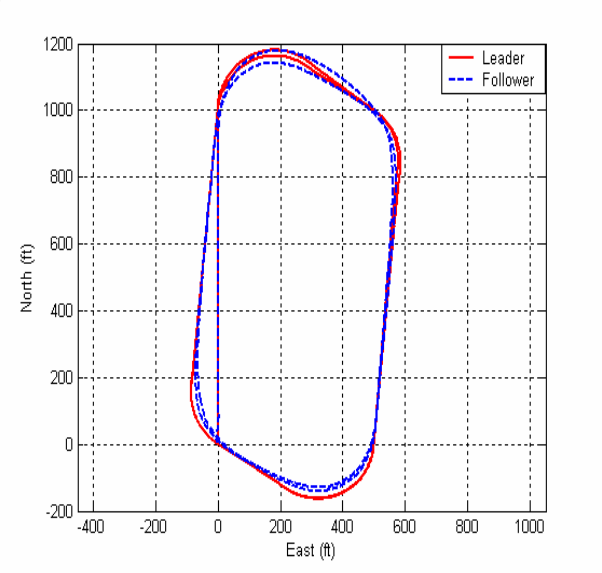


Figure 20. Formation oval trajectory.

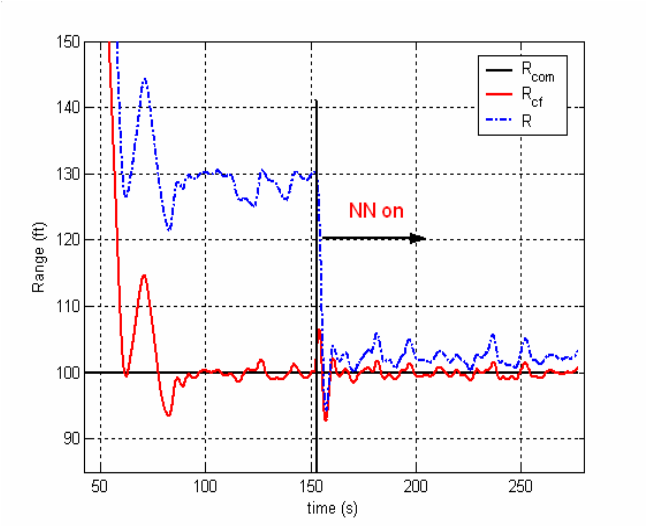


Figure 21. SITL simulation results-Range tracking performance for a racetrack maneuver.

The transient performance of the adaptive guidance controller is evaluated for a sudden stop maneuver in which the leader goes to a quick stop from a speed of 30 *ft/s*. The range tracking performance for the sudden stop maneuver is shown in Figures 22 and 23 with NN off and NN on, respectively. Even though the overshoot from the commanded range of 100 *ft* without the NN is lower compared to that with the NN, the maximum change in range during the transient is larger without the NN as compared to that with the NN. Without the NN, the range tracking transient goes from a steady state

of roughly 122 *ft* to the lowest value of roughly 85 *ft*, representing a change of approximately 37 *ft*. With the NN, the range tracking goes from roughly 95 *ft* to the lowest value of roughly 72 *ft*, representing a change of approximately 23 *ft*. Figure 24 compares the NN output with the leader line-of-sight (LOS) velocity, indicating that the NN is able to model the leader maneuver along the LOS direction very well.

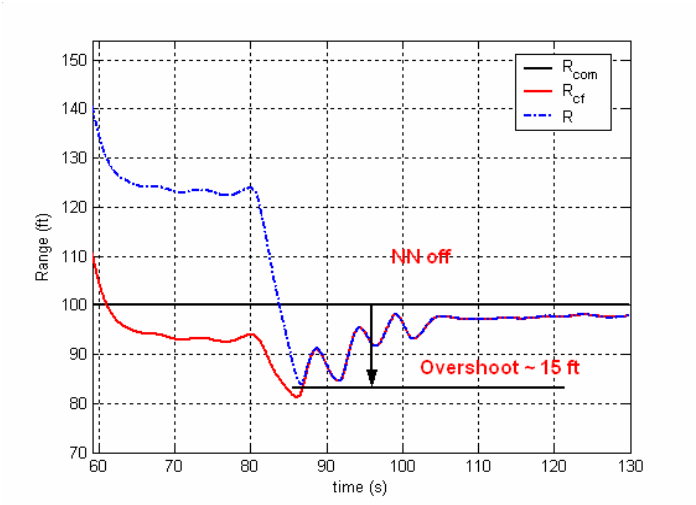


Figure 22. SITL simulation – Range tracking performance for a sudden stop maneuver (NN off).

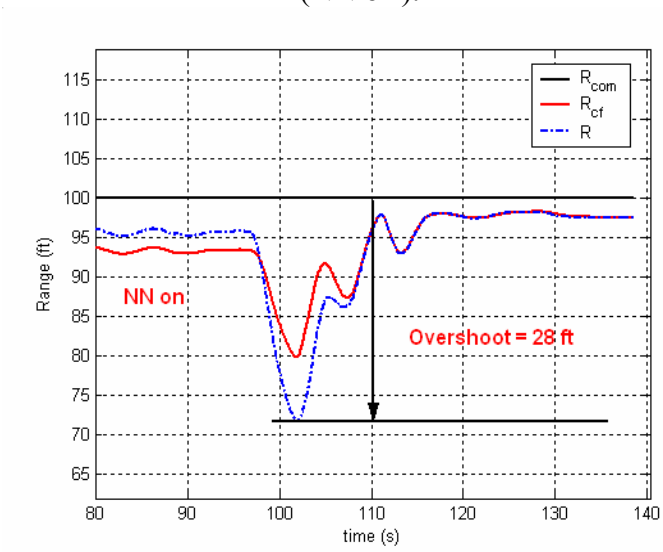


Figure 23. SITL simulation – Range tracking performance for a sudden stop maneuver (NN on).

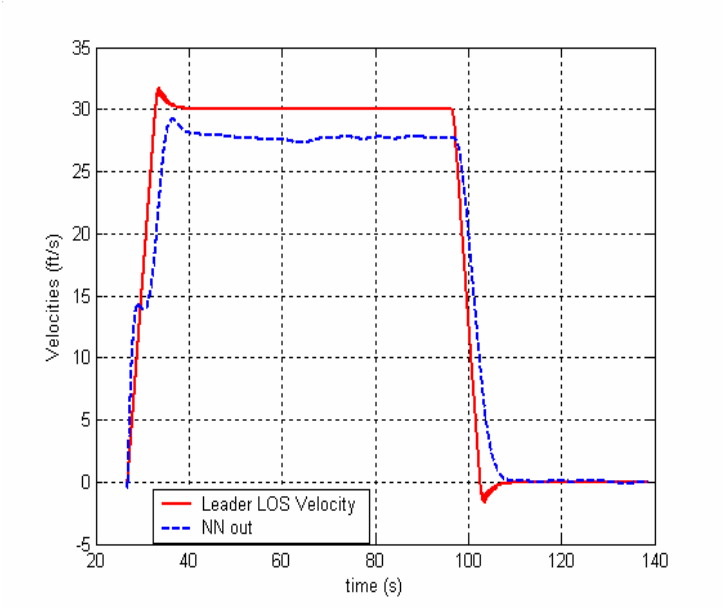


Figure 24. NN output vs. leader velocity along Line-of-Sight (LOS) (*ft/s*).

3.4 Flight Test Setup

In the flight test configuration, the sensor emulation, the vehicle model, and the actuator model are replaced by the actual vehicle. Also, during the flight test, the GCS software is run on one or more laptop computers and is used by human operators to interact with the onboard systems. The GCS is also equipped to read differential correction data from a GPS reference system and send it to the vehicle. The data link software provides connection between the GCS and onboard software.

Figure 25. Flight test setup

As shown in Figure 25, there are two vehicles used in the flight test: a ground vehicle and the GTMax. The ground vehicle is the leader or target vehicle instrumented with GPS and communication links. An instrumented fixed-wing UAV in the back of a van is used as the ground vehicle in the actual experiment. The van is driven around executing maneuvers of interest while being in communication with the ground control station (GCS). The GTMax is instrumented with integrated GPS/INS, onboard cameras for recording flight video, data recording and communication links. The GTMax is the follower vehicle in the flight test. Both the vehicles communicate with the GCS. The leader vehicle transmits GPS position, velocity and acceleration information to the GCS which then relays it to the follower. The flight computer onboard the follower compensates for the time delay associated with routing the leader data through the GCS and the slow update rate of GPS. The follower vehicle utilizes only the ***GPS position information*** of the leader vehicle for formation command tracking. The safety and performance of the GTMax is monitored from the GCS.

3.5 Flight Test Results

Before the flight evaluation experiment begins, the follower is put in hover at an altitude of 50 *ft* and the ground vehicle is driven to a horizontal distance of about 200 *ft* from the follower. The follower is commanded to maintain a horizontal distance (range) of 100 *ft* from the leader vehicle. Once the evaluation begins, the leader vehicle starts to execute a maneuver of interest. Simultaneously, leader data is communicated to the follower via the GCS and data recording begins onboard the follower. Then the follower vehicle is given an explicit command to climb to an altitude of 200 *ft* and maintain commanded horizontal range from the leader. Altitude is regulated independently to 200 *ft*. The leader vehicle performs a predetermined maneuver to verify the performance of the adaptive guidance law. The leader vehicle starts from rest and drives around in a racetrack pattern (oval trajectory) at 15 *ft/s*. Limits are imposed on maximum acceleration and heading rate of the leader vehicle. The experiments include switching on and off the adaptive component (NN on/ NN off) of the guidance law to demonstrate the effectiveness of adaptation.

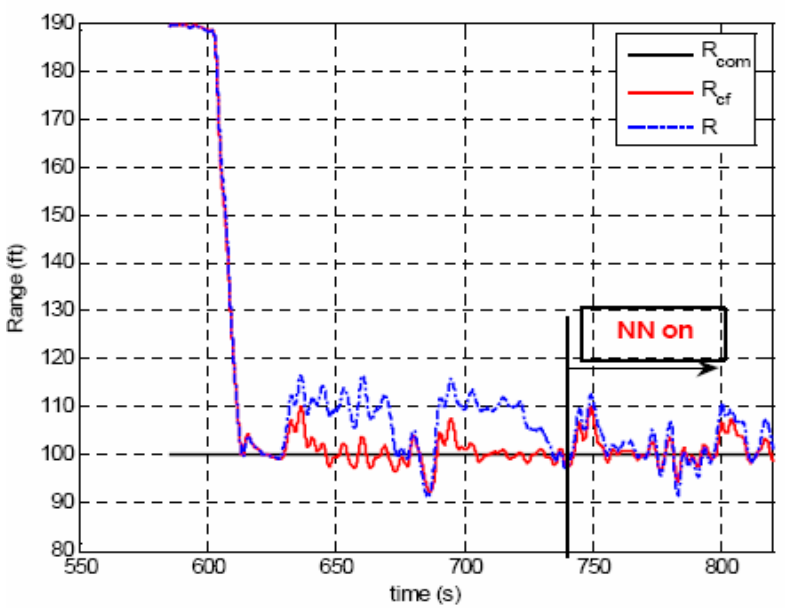


Figure 26. Flight test results - Range tracking performance for a racetrack maneuver.

Figure 26 shows the range tracking performance of the adaptive guidance controller for formation flight. R_{com} is range command (100 *ft* in this case), R_{cf} is hedged filtered range command and R is range response. The range tracking performance shows a steady-state error with NN off (time $\leq$ 739s). The one instance where the range comes close to the commanded value is when the ground vehicle slows down while turning in the loop. With the NN on (time $\geq$ 739s), the range tracking performance is generally improved.

Figure 27 shows the NN output and the leader velocity along the LOS. The latter is the uncertainty for the velocity guidance law. It is clear that as soon as the NN is switched on, the NN captures the uncertainty fairly accurately. Figure 28 shows the velocity command tracking. This plot shows that the commands issued by the guidance law are tracked accurately with the GTMax adaptive flight controller.

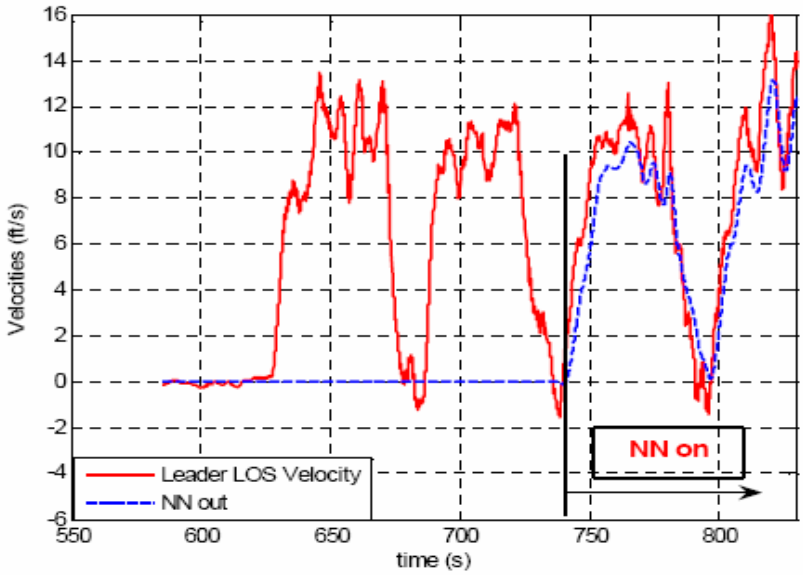


Figure 27. Flight test results - NN output vs. leader velocity along the LOS.

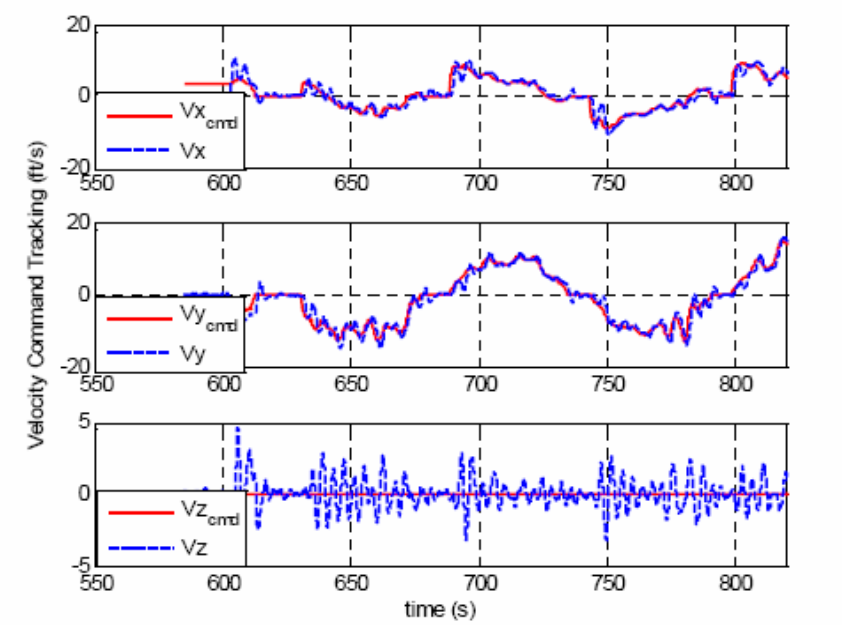


Figure 28. Flight test results - velocity command tracking for a racetrack maneuver.

The transient performance of the adaptive guidance controller was also evaluated using a series of sudden stop maneuvers by the ground vehicle. Figures 29 and 30 show the range tracking performance and the leader and the follower x- and y- velocities for the case of NN off. Figures 31 and 32 show similar results for the case of NN on. Note that for the case of NN on, due to a data overload problem onboard the vehicle, R_{cf} could not be recorded and hence not shown in Figure 31. From these flight test results, it is seen that the transient performance of the adaptive guidance controller is similar to that previously observed in the SITL simulations.

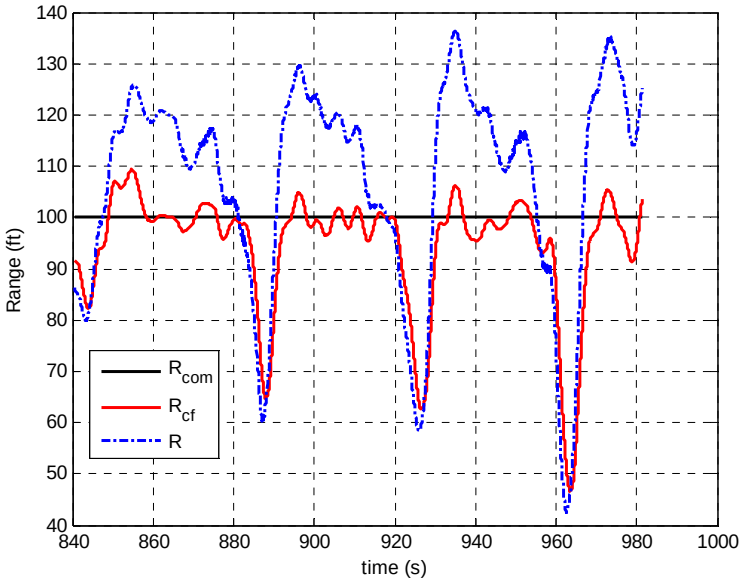


Figure 29. Flight test results – Range tracking performance for a series of sudden stop maneuvers (NN off).

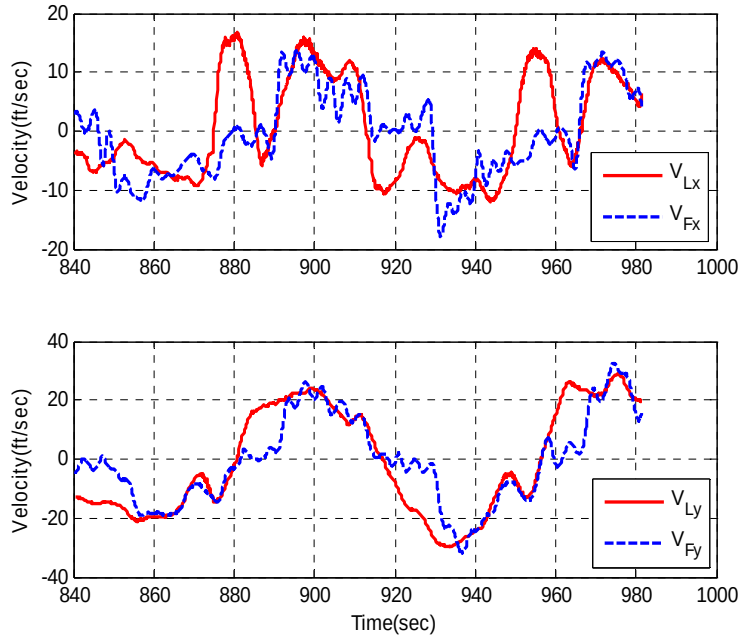


Figure 30. Flight test results – Leader and follower x- and y- velocities for a series of sudden stop maneuvers (NN off).

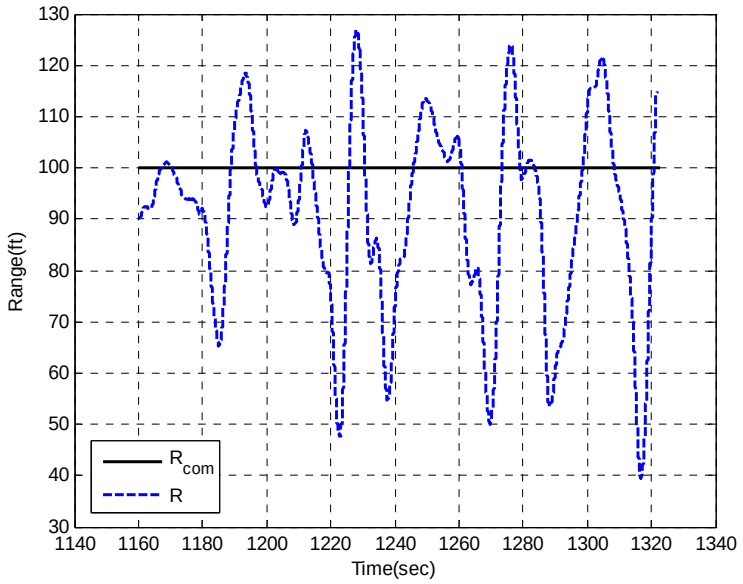


Figure 31. Flight test results – Range tracking performance for a series of sudden stop maneuvers (NN on).

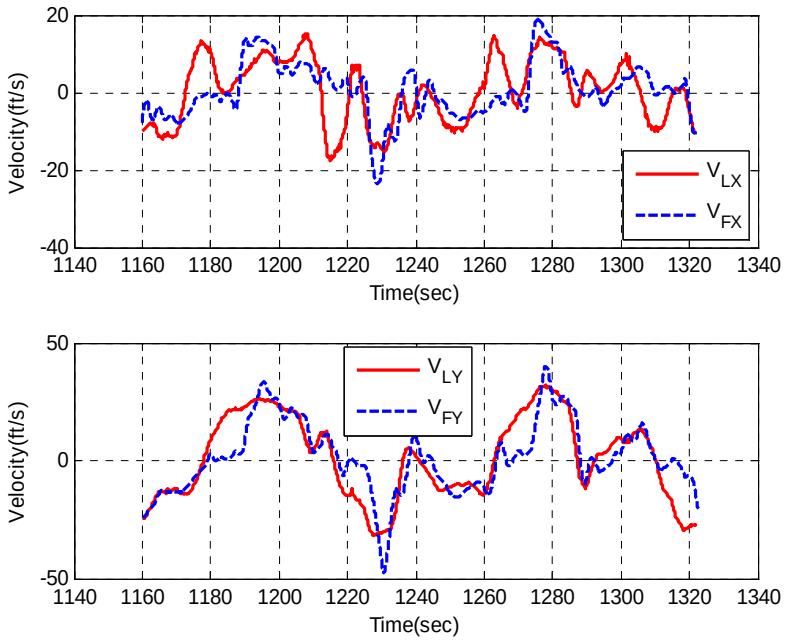


Figure 32. Flight test results – Leader and follower x- and y- velocities for a series of sudden stop maneuvers (NN on)

3.6 Chapter Summary

This chapter developed an adaptive guidance controller for autonomous formation flight. Both a velocity command guidance system and an acceleration command guidance system are developed and evaluated. The design of the adaptive guidance law is shown to be independent of the overall system architecture, and the integration of the guidance system with the adaptive autopilot of the Georgia Tech rotary wing UAV test bed (GTMax). In the design of the adaptive guidance law, the leader information except position is assumed to be unknown to a follower. Thus, the only measured information related to the leader is the LOS range and angle. Therefore, this technique can be applied to the Autonomous Formation Flight (AFF) control systems that use a passive sensing method.

The proposed AFF controller is evaluated using the software-in-the-loop simulations and flight test. Both results clearly show that the adaptive guidance control system is a promising solution for autonomous formation flight of UAVs. The successful flight evaluation is also important because the GTMax is a rotary wing platform and unique maneuvers, compared to fixed-wing vehicles.

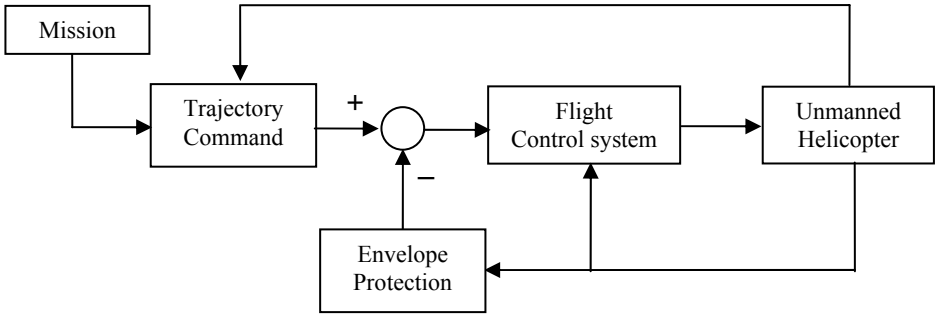
CHAPTER 4

MINIMUM-TIME APPROACH TO OBSTACLE AVOIDANCE

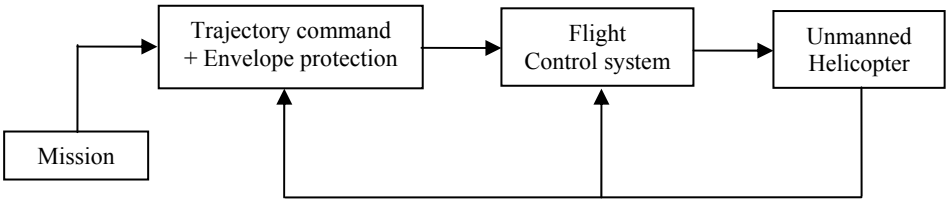
In this chapter, an integrated approach to combine obstacle avoidance and envelope protection for autonomous UAVs is proposed and developed. The method uses a minimum-time optimization formulation, and numerical solutions to the problem are obtained in form of acceleration commands.

4.1 Introduction

Unmanned aerial vehicles (UAVs) are expected to perform missions in a more efficient and less costly fashion than has historically been the case with manned vehicles. A related reason is that freeing machines from the limitations imposed by humans would increase their performance. Also, since UAVs are expected to operate in unknown and adversarial environments, issues on their safety need to be investigated extensively. Thus, the vehicle safety may be endangered in two different ways: external hazardous objects and internal performance limitations. Furthermore, certain mission tasks may result in aggressive maneuvering through obstacle fields, and it is imperative that the flight envelope of a UAV is protected to ensure safety and structural integrity. In addition, it is also desired that interruption of the original flight is minimized from a mission effectiveness point of view. The conflicting requirements of aggressive maneuvering for obstacle avoidance and the restricted maneuvering for envelope protection require new methods that combine obstacle avoidance and envelope protection into a unified framework.



(a) Command limiting for envelope protection



(b) Integrated trajectory command with envelope protection

Figure 33. Obstacle avoidance and envelope protection within UAV autonomy.

Figure 33 compares two different approaches to obstacle avoidance and envelope protection. Figure 33(a) shows the case where the trajectory command block and the envelope protection block work independently: Commands for obstacle avoidance are generated in the trajectory command block, whereas command limiting is done by the envelope protection block. Sometimes, these two blocks may compete with each other in order to achieve their objectives. This competition may result in a degradation of the mission performance, or cause the vehicle to run into obstacles. A more desirable alternative is to design guidance commands in consideration of the limit parameter response, so to say, an integrated approach as depicted in Figure 33(b). In the integrated approach, an answer to the following question is sought: *how one can determine guidance commands for accomplishing the overall mission objectives while simultaneously taking into account of both external and internal constraints in the*

presence of uncertainty? In order to answer this question, this chapter builds on a constrained optimal control approach.

4.2 PROBLEM FORMULATION

Let $\vec{X}, \vec{V}, \vec{a}$ be position, velocity and acceleration vectors, respectively, of the mass center of a UAV. The kinematics equations can be written for motion in the three-dimensional space, as

$$\dot{\vec{X}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \vec{V} \tag{56}$$

$$\dot{\vec{V}} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \vec{a} \tag{57}$$

where x, y, and z are inertial horizontal, lateral, and vertical positions, u, v, and w are inertial horizontal, lateral, and vertical velocities, and a_x, a_y, and a_z are inertial horizontal, lateral, and vertical accelerations. In addition, the UAV plus its flight controller can be approximated by a simplified model with the appropriate commands as inputs. Without any loss of generality of the approach, an acceleration command system for the UAV is assumed, and a first order model for the optimal controller formulation is used in this study.

$$\dot{\vec{a}} = \begin{bmatrix} \dot{a}_x \\ \dot{a}_y \\ \dot{a}_z \end{bmatrix} = \begin{bmatrix} \frac{1}{\tau_x}(-a_x + a_{x,c}) \\ \frac{1}{\tau_y}(-a_y + a_{y,c}) \\ \frac{1}{\tau_z}(-a_z + a_{z,c}) \end{bmatrix} \tag{58}$$

where $a_{x,c}$, $a_{y,c}$, and $a_{z,c}$ are horizontal, lateral, and vertical acceleration commands to be determined and τ_x , τ_y , and τ_z are time constants associated with the horizontal, lateral, and vertical channels of the acceleration command system.

Let us assume that $\vec{X}_{wp} = \left[x_{wp}\ y_{wp}\ z_{wp}\right]^T$ is given as a final destination point. A UAV is assumed to be flying toward the final waypoint. Let R_{ob} be a region of obstacle in the local frame. The region R_{ob} does not change with respect to time, i.e., a stationary obstacle. It is modeled as a rectangular 3-D object. The location of the obstacle is not known *a priori* to the vehicle. For collision avoidance, the vehicle should be prevented from entering the prohibited region surrounding an obstacle with pre-selected clearances. An obstacle detection system, for example a radar or a camera, is assumed to be used for obstacle detection during flight.

Once an obstacle is detected, then the integrated obstacle avoidance with envelope protection system is activated. The first thing this guidance system does is to locate a safe intermediate waypoint, $(x_{safe}, y_{safe}, z_{safe})$. This intermediate waypoint is determined from the shape of the detected obstacle and the selected margins of clearance from the obstacle, and not from the point of vehicle maneuverability. Then the integrated trajectory generation block in Figure 23(b) finds a way to reach the safe waypoint with minimal interruption to the original mission. For missions such as surveillance and reconnaissance, it is typically desirable to have the UAV to follow the original flight to the maximum extent, and hence, avoid obstacles with maximum agility. In order to use a full flight envelope in reaching the safe intermediate waypoint, a minimum-time problem with constraints can be formulated as follows:

$$J = t_f \tag{59}$$

subject to the equations of motion 56 ~ 58 and the following constraints. In equation 59, t_f represents the time elapsed during the obstacle avoidance maneuver.

Obstacle avoidance constraint:

$$\{x(t), y(t), z(t)\} \notin R_{ob} \text{ for } \forall t \in [0, t_f] \tag{60}$$

Limit parameter constraint:

$$y_{p,L} \le y_p(t) \le y_{p,U} \text{ for } \forall t \in [0, t_f] \tag{61}$$

where y_p is a limit parameter, and $y_{p,U}$ and $y_{p,L}$ are its upper and lower boundaries, respectively. The obstacle avoidance constraint in equation 61 can be converted into a safe way point (terminal constraint for the avoidance maneuver) constraint along with an implicit requirement that the horizontal velocity along the optimal avoidance maneuver must remain positive.

$$x(t_f) = x_{safe}, y(t_f) = y_{safe}, z(t_f) = z_{safe} \tag{62}$$

$$u(t) \ge 0 \text{ for } \forall t \in [0, t_f] \tag{63}$$

Furthermore, by imposing zero lateral and vertical velocity constraints at the safe way point, i.e.,

$$v(t_f) = 0, w(t_f) = 0 \tag{64}$$

excessive overshoot above the obstacle should be avoided.

Once a time-optimal solution is obtained, the algorithm determines whether the avoidance maneuver should be initiated right away according to the obtained acceleration

command solution. If the time-optimal solution requires positive acceleration towards an obstacle at the instance of obstacle detection, this situation is termed as a '*warning state*.' In this state, the vehicle can maintain its original flight path as there is room for it to accelerate towards the obstacle. In this case, an obstacle avoidance maneuver is initiated at a point when the optimal horizontal acceleration command becomes zero. If at the instance of obstacle detection, the time-optimal solution requires that the UAV has to decelerate while meeting the implicit requirement that the optimal horizontal velocity is positive along the entire obstacle avoidance trajectory, this condition is termed as '*safe avoidance state.*' In this case, the avoidance maneuver is initiated right away. If the optimal solution does not meet the implicit requirement that the horizontal velocity is positive along the avoidance trajectory, such a solution violates the obstacle constraint even though it meets the limit parameter constraint. This condition is termed as '*unsafe avoidance state*.' In this case, the vehicle must slow down to reach a 'safe avoidance state' before it initiates an avoidance maneuver. Figure 34 shows a schematic representation of safety states for obstacle avoidance [65].

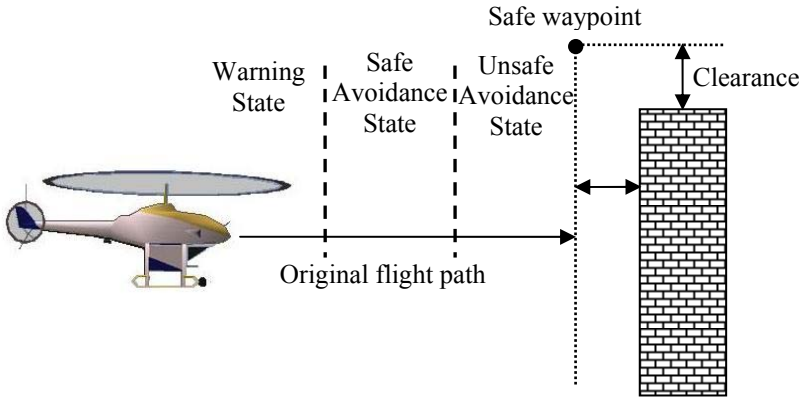


Figure 34. Safety states for obstacle avoidance.

4.3 NONLINEAR TRAJECTORY GENERATION FOR REAL-TIME OPTIMAL SOLUTION

The classical collocation method [28] is one among the many different methods to find solutions for optimal control problems. However, it is difficult to use this approach for real-time application since it is based on discretization which results in large number of unknowns to be solved simultaneously. Nonlinear trajectory generator (NTG) [60][61]is a software package that can be used to find optimal solutions for differentially flat systems in real-time. Differential flat system, by definition [21], can be completely described using a few set of variables and their derivatives. These variables are referred to as flat outputs of the systems. A necessary condition for the existence of such an output can be found in Reference [21].

There are three primary components to the NTG methodology. The first is to determine flat outputs such that system equations can be mapped to a lower dimensional output space. Once flat outputs are determined, the cost to be minimized and the constraints can be mapped to the output space. The idea is that it will be easier as well as computationally more efficient to solve a lower dimensional problem. In most case, it is desirable to find an output $z = \lfloor z_1, z_2, \cdots, z_q \rfloor$ in form of

$$z = P\left(x, u, u^{(1)}, \cdots, u^{(r)}\right) \tag{65}$$

Such that $\left(x(t), u(t)\right)$ can be determined completely from

$$\left(x(t), u(t)\right) = Q\left(z, z^{(1)}, \cdots, z^{(s)}\right) \tag{66}$$

where $x^{(i)}$ denotes the *i*th time derivative of *x*.

The second is to parameterize the flat outputs in terms of B-spline curves [7]. The order and multiplicity of B-spline basis functions are chosen based on smoothness

condition for individual variables. The choice of order, multiplicity, and intervals fixes the set of basis functions that can be used to describe the flat outputs and other variables. The flat outputs are written in terms of finite dimensional B-spline curves as

$$\begin{aligned} z_1 &= \sum_{i=1}^{p1} B_{i,k1}(t) C_i^{\,1} \\ z_2 &= \sum_{i=1}^{p2} B_{i,k2}(t) C_i^{\,2} \\ &\vdots \\ z_q &= \sum_{i=1}^{pq} B_{i,kq}(t) C_i^{\,q} \\ &\text{and } p_j = l_j\left(k_j - m_j\right) + m_j \end{aligned} \tag{67}$$

where $B_{i,k_j}(t)$ is the B-spline basis function for the flat output z_j with order k_j, $C_i^{\,j}$ are the coefficients of the B-spline, l_j is the number of intervals, and m_j is the number of smoothness conditions. The coefficients of the B-spline basis functions will be determined using sequential quadratic programming.

Now, the original optimization problem is converted to nonlinear programming problem in form of

$$\begin{aligned} &\min F(y) \\ &\text{Subject to } b_l \le c(y) \le b_u \end{aligned} \tag{68}$$

where

$$y = \left(C_1^{\,1}, \cdots, C_{p1}^{\,1}, C_1^{\,2}, \cdots, C_{p2}^{\,2}, \cdots, C_1^{\,q}, \cdots, C_{pq}^{\,q}\right) \in \mathbf{R}^M \qquad M = \sum_{i=1}^{q} p_i \tag{69}$$

and $F(y)$ is the discrete approximation in output space to the objective function. The lower and upper bounds for the constraints are denoted by b_l and b_u. Finally, the sequential quadratic programming package NPSOL [24] is used as the nonlinear programming solver in NTG to find the optimal set of these coefficients that minimize

the given cost function while satisfying a selected set of constraints. Reference [72] shows an example of the NTG application to the real-time control of the Caltech ducted fan.

4.4 METHOD OF SOLUTION

It is expected that the aggressive maneuvers needed for obstacle avoidance have a tendency to drive limit parameters to their boundaries. Thus, in order to maintain the vehicle within the safe operational envelope, the aggressiveness of the avoidance maneuver should be restricted by the limit parameter dynamics, which means that admissible control inputs are also restricted by limit parameter dynamics.

Assume that a constraint on limit parameter is given by equation 58. The limit parameter dynamics in general can be expressed as

$$y_p^{(r)} = h\left(y_p, y_p^{(1)}, \cdots, y_p^{(r-1)}, \vec{X}, \vec{V}, \vec{a}, a_{x,c}, a_{y,c}, a_{z,c}\right) \tag{70}$$

where $h(\cdot)$ represents the nonlinear limit parameter dynamics, and r is the relative degree of the limit parameter dynamics. By definition [44], relative degree means how many differentiations with respect to time of the limit parameter are required for control variables to explicitly appear as in equation 70. If $r = 0$, then the relationship between control inputs and the limit parameter can be expressed algebraically. Hence, limit boundaries may be translated to control variable inequality constraints (CVIC). For example, by assuming that acceleration commands can be achieved instantaneously (neglecting equation 58), the time-optimal control problem with load factor constraints is transformed to a problem with CVIC [64][66]. For this case, after defining the Hamiltonian of the system, an optimal solution can be determined using the Pontryagin's minimum principle [9][36][55].

$$H\left(\vec{X}^*, \vec{V}^*, \vec{\lambda}^*, \vec{a}^*\right) \leq H\left(\vec{X}^*, \vec{V}^*, \vec{\lambda}^*, \vec{a}\right) \text{ for } \forall \vec{a}(t) \in \mathbf{A} \tag{71}$$

where H represents the Hamiltonian, and * represents values along the optimal trajectory. In addition, $\vec{\lambda}$ means a co-state vector, and $\mathbf{A}$ means a set of admissible control inputs, which is given by

$$\mathbf{A} = \left\{ \left(a_{x,c}, a_{y,c}, a_{z,c} \right) \middle| \, y_{p,L} \le h\left(\vec{X}, \vec{V}, \vec{a} \right) \le y_{p,U} \right\} \tag{72}$$

In addition, since a final time is free, the transversality condition should also be satisfied as follows:

$$H\left(t_f \right) = 0 \tag{73}$$

Then, by imposing the terminal constraints in equation 62 and 64, an optimal solution for the case of envelope limits of relative degree 0 can be obtained from equations 71 ~ 73.

On the other hand, for the case of a limit parameter of relative degree $r \ge 1$, the limit parameter and its time-derivatives can be considered as additional state variables, and the limit parameter constraints become state variable inequality constraints (SVIC) [10]. In other words, an augmented state vector, ξ, can be defined as

$$\xi = \left[x, y, z, u, v, w, a_x, a_y, a_z, y_p, y_p^{(1)}, \cdots, y_p^{(r-1)} \right]^T \tag{74}$$

Then, the augmented system of equations for the optimal control problem become

$$\dot{\xi} = \begin{bmatrix} u \\ v \\ w \\ a_x \\ a_y \\ a_z \\ \frac{1}{\tau_x}\left(-a_x + a_{x,c}\right) \\ \frac{1}{\tau_y}\left(-a_y + a_{y,c}\right) \\ \frac{1}{\tau_z}\left(-a_z + a_{z,c}\right) \\ y_p^{(1)} \\ y_p^{(2)} \\ \vdots \\ h(\xi) \end{bmatrix} \tag{75}$$

When using the nonlinear trajectory generator for solving the optimal control problem numerically, it is noted that the order and multiplicity of B-spline basis functions should be chosen according to relative degree of the limit parameter. For example, in case of the limit parameter of relative degree greater than 1, the B-spline basis function for the limit parameter should be continuously differentiable at least *r*-1 times since the limit parameter must have at least *r*-1 continuous derivatives. In equation 75, the flat outputs are horizontal position *x*, lateral position *y*, vertical position *z*, limit parameter y_p, and final time t_f. As explained in Section 4.2, in order to further reduce the number of computations needed for obtaining a solution using NTG, the state inequality constraints for obstacle avoidance are accounted for by a combination of explicit terminal state constraints given by equation 62 and an implicit constraint that the resulting solution does not require negative horizontal velocity during any part of the avoidance maneuver given by equation 63.

4.3.1 Nominal solution case

The safety state of the vehicle is determined based on the optimal horizontal acceleration command and the optimal horizontal velocity profile. 1) If $a_{x,c}{}^{*}(t) > 0$, this is the '*warning state*,' and there is room to maintain the original flight path, thus implicitly meeting the original goal that the vehicle should maintain its original flight path to the maximum extent possible. 2) When $a_{x,c}{}^{*}(t) = 0$, the elapsed time for the avoidance maneuver can be shown to be the absolute minimum based on the fact that lateral and/or vertical acceleration commands can be fully used without sharing their effect on limit parameter dynamics with horizontal acceleration command. 3) If $a_{x,c}{}^{*}(t)$ is negative and $u^{*}(t) \geq 0$ for $\forall\, t \in [0, t_f]$, then the vehicle is in the '*safe avoidance state*,' and should immediately start to execute the avoidance maneuver. These three cases are categorized as nominal solution cases.

4.3.2 Unsafe solution case

In some cases, for example, if the obstacle is detected too late or the vehicle velocity is too high, then the time-optimal solution from the NTG may violate the implicit requirement that the horizontal velocity is non-negative throughout the avoidance maneuver. If $a_{x,c}{}^{*}(t) < 0$ at the time instant an obstacle is detected, and there exists $t_1 \in [0, t_f]$ such that $u^{*}(t_1) < 0$, then it represents an unsafe avoidance case. Since this solution implies that the vehicle is expected to enter the obstacle region and come back to the safe waypoint, an alternative strategy is required to avoid the obstacle. In such a case, the vehicle needs to execute a horizontal deceleration maneuver prior to initiating an avoidance maneuver. In this study, the deceleration maneuver is assumed to be performed at a pre-selected deceleration rate, typically at the maximum permissible deceleration.

The horizontal deceleration is continued until such point when a safe time-optimal avoidance solution is obtained from the NTG. Another alternative is to continue the deceleration until a point when $a_{x,c}^{*} = 0$. These two alternatives are pursued and evaluated further in this study. As a worst case scenario, the vehicle may come to hover, and climb vertically. This is a possible solution especially for rotary wing UAVs.

4.3.3 Multiple safe waypoints case

The proposed algorithm may also be used to handle the case of multiple safe waypoints in a three-dimensional space. Figure 35 depicts an example of multiple safe waypoints. Since safe waypoints are selected by adding some clearance to obstacle configuration, it is possible that there exist multiple safe waypoints in the three-dimensional space. The three-dimensional space is divided into multiple two-dimensional planes and a safe waypoint is selected in each plane. The proposed algorithm solves the time-optimal control problem for each of the selected safe waypoints and selects the solution that will permit maintaining the original flight to a greater extent.

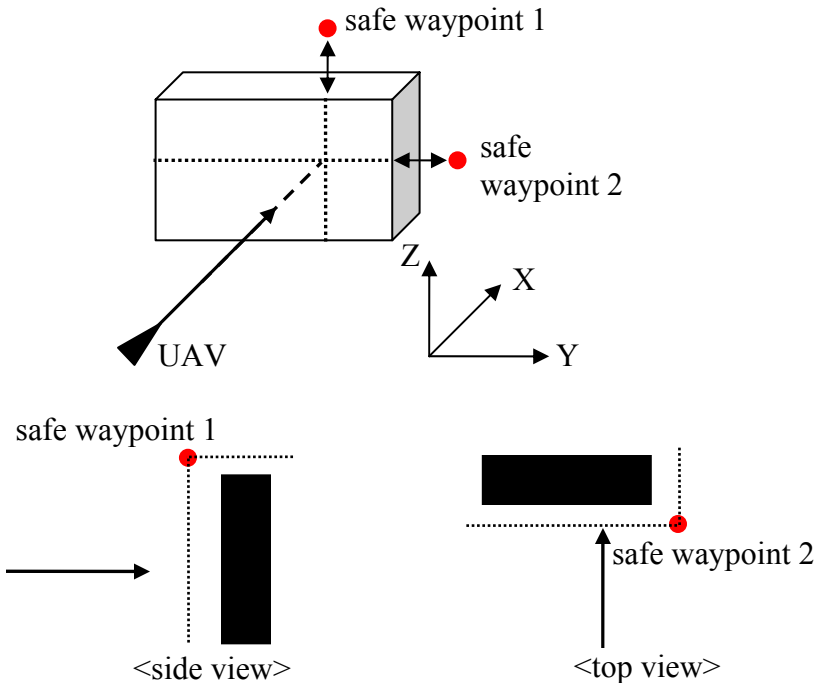


Figure 35. Example of multiple safe waypoints.

Of the multiple optimal solutions, if there is at least one solution that requires a positive horizontal acceleration command, then the UAV maintains its original flight path and executes an avoidance maneuver in that plane when the horizontal acceleration command required from the time-optimal solution becomes zero. If both solutions require horizontal acceleration commands less than or equal to zero, then the valid solution that corresponding to lower of the two minimum times is selected by the algorithm and the avoidance maneuver in the corresponding plane is executed.

4.3.4 Unidentified obstacle size case

So far, it is assumed that the obstacle size is fully identified with an obstacle detection device. However, it is possible that the obstacle detection device cannot figure out the entire size of the obstacle for some reasons, for example, the performance limit of the obstacle detection device, bad weather conditions, etc. As a matter of fact, approaches to this situation are strongly related with the mechanism of the obstacle detection device

and the search algorithm. When the obstacle size is not fully known, the vehicle may try to figure out the obstacle configuration. Alternatively, by assuming that the obstacle has infinite size, the vehicle immediately starts avoidance maneuvers [75].

For this case, this chapter focuses on the design of the guidance algorithm. Thus, the proposed method should determine obstacle avoidance maneuvers based on the obstacle information obtained until that instance. Also, if more obstacle information becomes available during the maneuver, the safe waypoint changes accordingly, and the proposed method re-optimizes obstacle avoidance maneuvers with respect to a changed safe waypoint. This process may result in a rapid deceleration along the horizontal direction.

4.5 Software-in-the-loop Simulation Results

The proposed algorithm is evaluated in simulations using a nonlinear flight dynamic model of a GTMax-like rotary wing test bed within the Georgia Tech UAV simulation tool (GUST) [40][41]. This model is identical to the GTMax except the engine power. A block diagram representation of the flight controller used in this study is shown in Figure 36.

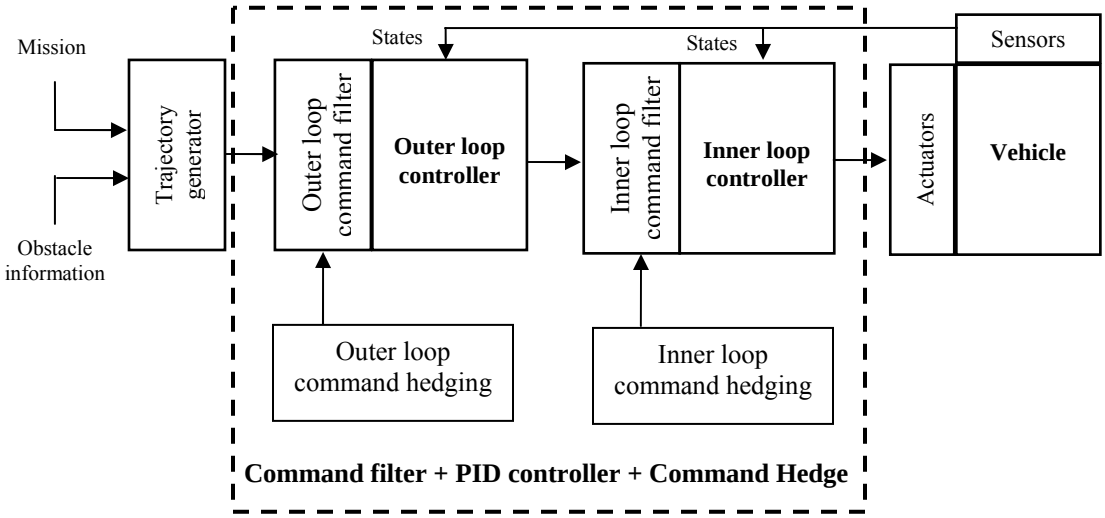


Figure 36. Flight controller architecture used in simulation evaluation.

The first component, the trajectory generator in Figure 36, provides the commands to the flight controller. In the absence of any obstacles, the commands from a trajectory generator are fed into the flight controller in order to follow a given set of waypoints from a mission planner. If an onboard obstacle detection device finds an obstacle along the original path, the trajectory generator commands are modified using the proposed algorithm. The default flight controller represented as dashed box in Figure 36 is a proportional-derivative-integral (PID) trajectory following controller. Since PID autopilot controllers are more conventional than NN-based autopilot controllers, performance evaluation is performed using a PID controller. Since this flight controller requires position and velocity commands as well as acceleration commands, we need to integrate acceleration commands over time in order to get position and velocity commands. The values of the trajectory controller parameters used in simulations for this study are the same as presented in Table 3 in Chapter 3. Accordingly, the time constants τ_x, τ_y, and τ_z in equation 58 for the approximate dynamics of the vehicle plus the flight controller used for optimal controller solutions are set at 0.5 sec, 0.4 sec , and 0.33 sec, respectively.

4.5.1 Load factor limiting

Consider a limit on the aerodynamic load factor given by

$$N_z = \frac{\sqrt{a_x{}^2 + a_y{}^2 + (a_z + g)^2}}{g} \leq N_{z,\max} \tag{76}$$

Two different cases are considered in this thesis to consider load factor limiting.

Case of neglected UAV + autopilot dynamics

First, by assuming that commanded acceleration can be achieved instantaneously, equation 58 can be excluded during the optimization process. Then, control variables become vehicle accelerations, which means that the relative degree of load factor becomes zero. Without loss of generality, it is assumed that a UAV is restricted to move in the vertical plane. Let us define the Hamiltonian as follows:

$$H = 1 + \lambda_1 u + \lambda_2 w + \lambda_3 a_x + \lambda_4 a_z \tag{77}$$

The Euler-Lagrange equations for the co-states can be obtained as follows:

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x} = 0, \;\; \lambda_1 = c_1 \tag{78}$$

$$\dot{\lambda}_2 = -\frac{\partial H}{\partial z} = 0, \;\; \lambda_2 = c_2 \tag{79}$$

$$\dot{\lambda}_3 = -\frac{\partial H}{\partial u} = -\lambda_1, \;\; \lambda_3 = -c_1 t + c_3 \tag{80}$$

$$\dot{\lambda}_4 = -\frac{\partial H}{\partial w} = -\lambda_2, \;\; \lambda_4 = -c_2 t + c_4 \tag{81}$$

where $c_i, \;\; i = 1, 2, 3, 4$ are constants. Since the final position and the final vertical speed are fixed, and the final horizontal speed is free, the final conditions of the co-states are written as

$$\lambda_1(t_f) = \text{free},\ \lambda_2(t_f) = \text{free},\ \lambda_3(t_f) = 0,\ \text{and}\ \lambda_4(t_f) = \text{free} \tag{82}$$

Finally, from equation 64, the co-states are determined as follows:

$$\lambda_1 = c_1,\ \lambda_2 = c_2,\ \lambda_3 = c_1(t_f - t),\ \lambda_4 = c_2(t_f - t) + c_4 \tag{83}$$

In equation 83, three constant parameters, c_1, c_2, and c_4 are still undetermined, and they depend on the initial states $[x_0, z_0, u_0, w_0]$. It can be inferred that the sign of λ_3 does not change while that of λ_4 may change at most once. From equations 71 and 72, the Pontryagins's minimum principle can be expressed by

$$c_1(t_f - t)a_x^* + \{c_2(t_f - t) + c_4\}a_z^* \le c_1(t_f - t)a_x + \{c_2(t_f - t) + c_4\}a_z \tag{84}$$

$$\text{for}\ \forall \vec{a}(t) \in \mathbf{A} = \left\{ (a_x, a_z) \,\middle|\, \frac{\sqrt{a_x^{\,2} + (a_z + g)^2}}{g} \le N_{z,\max} \right\} \tag{85}$$

Since an optimal solution needs to minimize the value of $c_1(t_f - t)a_x + \{c_2(t_f - t) + c_4\}a_z$ given in equation 84 among elements of set $\mathbf{A}$, it can be represented by

$$a_x^*(t) = N_{z,\max} g \cos\alpha(t),\ \ a_z^*(t) = N_{z,\max} g \sin\alpha(t) - g \tag{86}$$

where $\alpha(t)$ represents the direction of acceleration command vector, which satisfies

$$\alpha(t) = \text{atan2}(-c_2(t_f - t) - c_4, -c_1(t_f - t)) \tag{87}$$

Equation 87 means that the optimal direction $\alpha(t)$ is opposite to the direction of the vector $(c_1(t_f - t),\ c_2(t_f - t) + c_4)$.

If the optimal solution is expressed by equations 86 and 87, the remaining unknowns are c_1, c_2, c_4, and t_f. These unknowns can be solved using the conditions at the final time as follows:

$$z(t_f) = z_0 + \int_0^{t_f} \left(w_0 + \int_0^{\tau} a_z^*(s) ds \right) d\tau = z_{safe} \tag{88}$$

$$x(t_f) = x_0 + \int_0^{t_f} \left(u_0 + \int_0^{\tau} a_x^*(s) ds \right) d\tau = x_{safe} \tag{89}$$

$$w(t_f) = w_0 + \int_0^{t_f} a_z^*(t) dt = 0 \tag{90}$$

$$H(t_f) = 1 + c_1 \left(u_0 + \int_0^{t_f} a_x^*(t) dt \right) u(t_f) + c_4 a_z^*(t_f) = 0 \tag{91}$$

Equations 88 ~ 90 represent final time constraints and equation 91 means the transversality condition at the final time.

Case of 1st order representation of UAV + autopilot dynamics

With the assumed form of the acceleration command system of equation 58, the relative degree of load factor limit parameter becomes one. Now, by augmenting the state vector with load factor as an additional state, the load factor constraint of equation 61 can be treated as a state variable inequality constraint within NTG. Table 4 lists details of the B-spline representations of flat outputs used within NTG.

Table 4. Details of flat outputs within NTG-load factor limiting.

Flat outputs	intervals	order	multiplicity	# of coeffi.
x	5	6	4	14
z	5	6	4	14
N_z	5	2	1	6
t_f	1	1	0	1

The proposed approach is applied to the case of obstacle avoidance in the vertical plane first. The UAV is assumed to be flying horizontally at a constant speed of 30 *ft/s* toward the destination waypoint, and detects an obstacle along the flight path. The region of the obstacle is modeled as a 3-D object with $R_{ob} = \{(700,0,0),(800,0,200)\}$, where the first point is the lower left corner and the second point is the upper right corner of the assumed rectangular 3-D obstacle. The location of the obstacle is not known to the UAV until it is detected by the obstacle detection system. The proposed approach for obstacle avoidance with a load factor constraint is activated at this moment, and it provides acceleration commands to the flight controller. Table 5 lists the values of flight and obstacle parameters used in simulations for this study.

Table 5. Parameter values used for obstacle avoidance with load factor limit simulations

Initial position of UAV (ft)	(0, 0, 30)
Destination waypoint (ft)	(1500, 0,30)
Initial Speed (ft/sec)	30/50
Maximum load factor	1.2/1.5/1.8
Maximum detection range (ft)	400
Field of view	60˚

The GUST simulation results of the vehicle trajectory response, the horizontal and vertical components of velocity response and the load factor response for the case of obstacle avoidance in the vertical plane with load factor limiting are shown in Figures 37 through 46, respectively. Simulation results show the cases of neglected UAV plus autopilot dynamics with and without the vertical velocity constraint at the safe waypoint.

Figures 37 and 38 show trajectory responses in the vertical plane for the cases without and with the vertical velocity constraint at the safe way point, respectively. When the vehicle detects an obstacle and predicts a collision when it reaches the point at (300,

0, 30), a safe waypoint is obtained by a pre-selected clearance from the upper left corner. At this instance, the time-optimal solution requires acceleration towards the obstacle, thus the safety state is the warning state. Therefore, the vehicle does not start the avoidance maneuver, and it maintains the original flight route. In Figure 37, when the vehicle reaches the point near at (540, 0, 30), an optimal acceleration along the horizontal direction becomes zero. At this point, the vehicle enters the safe avoidance state, and it starts the avoidance maneuver. In Figure 38, for the case with zero vertical velocity constraint at the safe way point, the vehicle should initiate avoidance maneuver earlier (roughly 15 *ft* in this example) than for the case without the zero vertical velocity constraint. As shown in Figure 37 for the case when there is no vertical velocity constraint at the safe way point, the vehicle continues the climb beyond the safe waypoint before it begins a descent flight to get to the end waypoint. This is caused by the fact that the maximum acceleration command permissible from the load factor constraint is applied which results in non-zero climb rate when the vehicle reaches the safe waypoint. On the other hand, as shown in Figure 38 for the case of zero vertical velocity constraint at the safe way point, the overshoot after reaching the safe waypoint is very much removed.

Figures 39 and 40 show comparison of velocity changes without and with a velocity constraint. In Figure 39, the vehicle still climbs after reaching the safe waypoint since the vertical velocity is positive. This causes a large amount of overshoot in the vertical position. In Figure 40, the vertical speed returns back to zero when the vehicle reaches the safe waypoint, and stays at the same velocity until the UAV becomes clear from the obstacle. Figures 41 and 42 show load factor changes without and with a final velocity constraint, respectively. In both cases, the maximum load factor is used during the avoidance maneuver.

In Figures 43 and 44, the trajectory responses are compared for different load factor limits without and with the velocity constraint. In both cases, it is noticed that with

a higher load factor limit, the vehicle can initiate obstacle avoidance maneuver later and maintain the original flight path longer. This is caused by the fact that a higher load factor limit allows the vehicle to maneuver more aggressively. However, the overshoot after reaching the safe waypoint also increases with a higher load factor limit as shown in Figure 43. Simulation results are summarized in Tables 6 and 7. For same load factor limit, the vehicle without the vertical velocity constraint reaches the safe waypoint faster. Figures 45 and 46 present a trajectory response comparison for different initial speeds without and with the velocity constraint. The UAV is assumed to be flying at 30 *ft/s* or 50 *ft/s*, and 1.5 is imposed as the load factor limit for both cases. It is seen that with a faster initial speed, the vehicle should start obstacle avoidance maneuver earlier. It is also shown that the vertical climb beyond the safe waypoint increases when the vehicle is originally flying faster. Tables 8 and 9 show simulation results for different initial speeds.

In Figures 39 and 40, it is interesting to notice that even though obstacle avoidance maneuver is initiated when the horizontal acceleration command becomes zero, the horizontal velocity decreases by 10 *ft/s* during the maneuver. This can be explained by the fact that the UAV plus autopilot system dynamics are neglected during the guidance system design. In fact, this degradation results from the assumption that commanded acceleration can be achieved instantaneously.

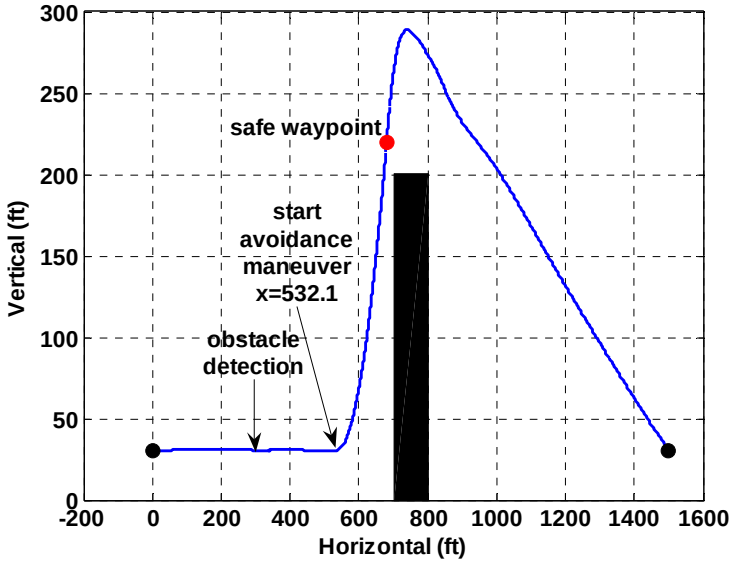


Figure 37. Trajectory response – neglected subsystems w/o zero vertical velocity constraint

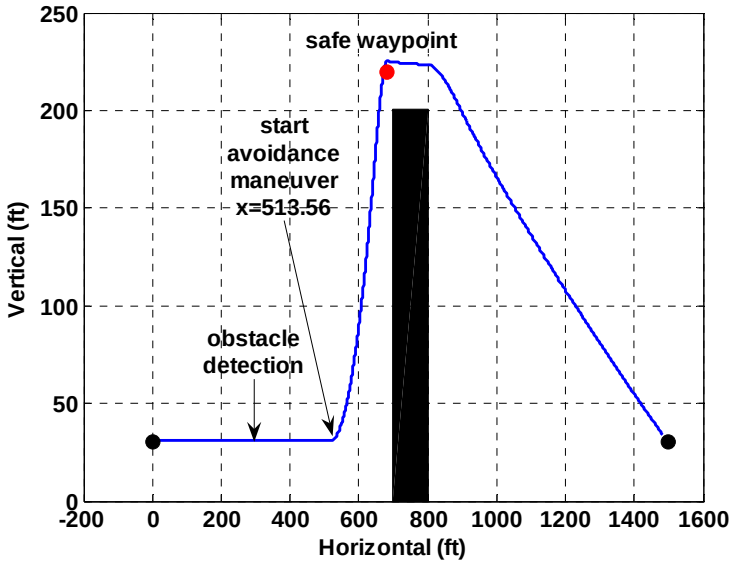


Figure 38. Trajectory response – neglected subsystems with zero vertical velocity constraint

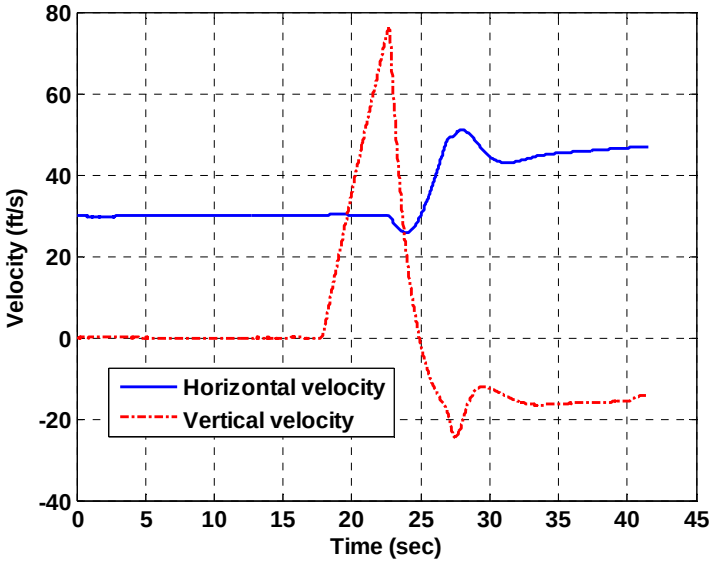


Figure 39. Velocity response – neglected subsystems w/o zero vertical velocity constraint

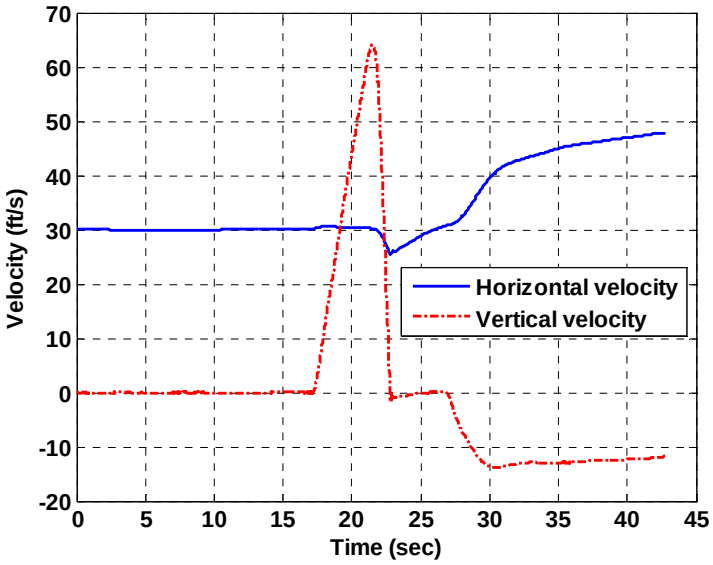


Figure 40. Velocity response – neglected subsystems with zero vertical velocity constraint

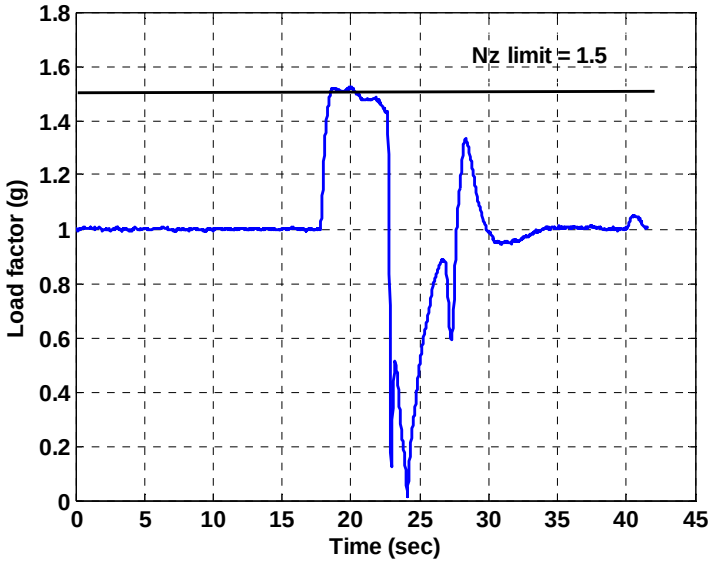


Figure 41. Load factor response – neglected subsystems w/o zero vertical velocity constraint

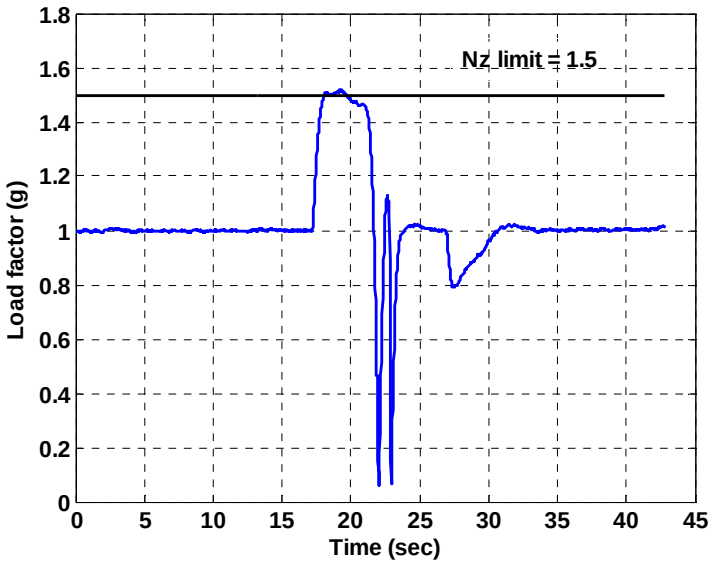


Figure 42. Load factor response – neglected subsystems with zero vertical velocity constraint

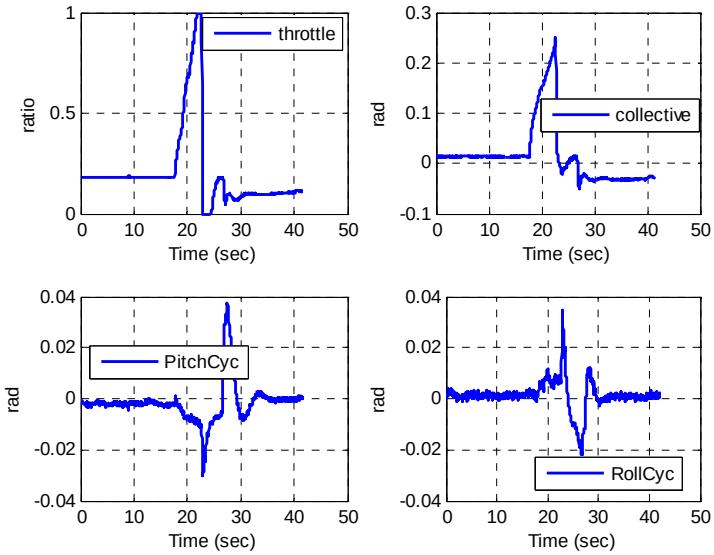


Figure 43. Control response – neglected subsystems w/o zero vertical velocity constraint

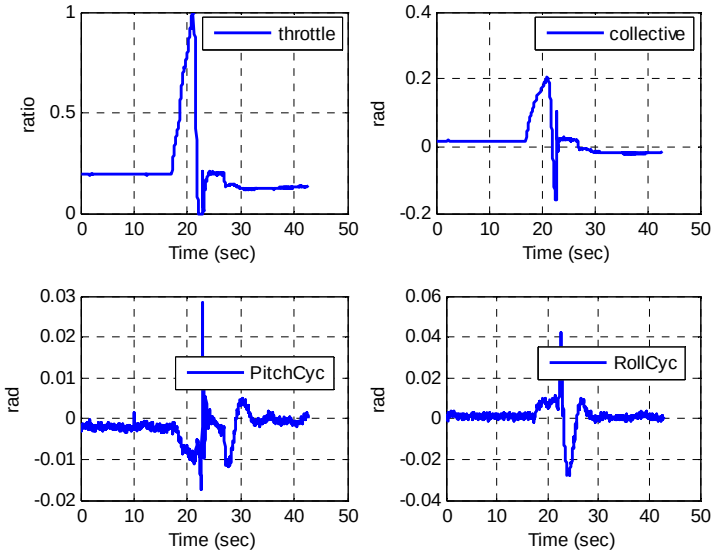


Figure 44. Control response – neglected subsystems with zero vertical velocity constraint

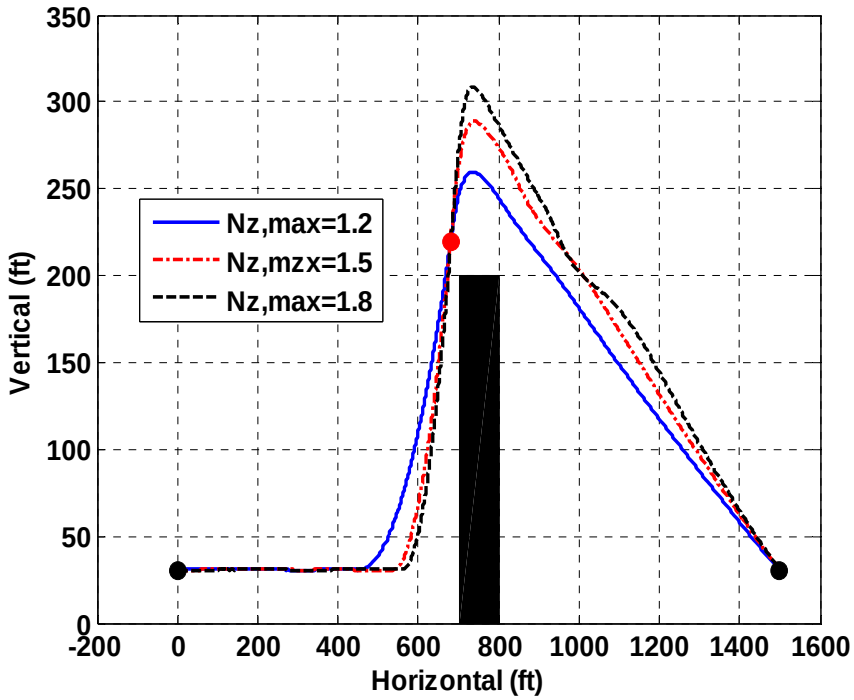


Figure 45. Trajectory response comparison for different load factor limits – neglected subsystems w/o vertical velocity constraint

Table 6. Simulation result comparison for different load factor limits – neglected subsystems w/o vertical velocity constraint

	Nz,max=1.2	Nz,max=1.5	Nz,max=1.8
Starting point (ft)	449.6	532.1	556.2
Time elapsed (sec)	7.66	4.99	4.12
Overshoot (ft)	40	70	90

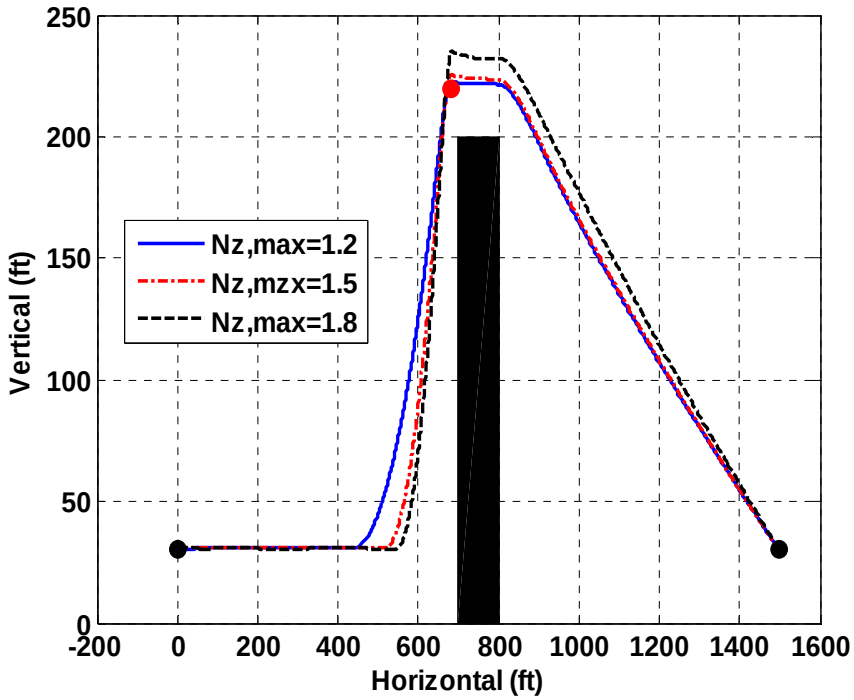


Figure 46. Trajectory response comparison for different load factor limits – neglected subsystems with vertical velocity constraint

Table 7. Simulation result comparison for different load factor limits – neglected subsystems with vertical velocity constraint

	Nz,max=1.2	Nz,max=1.5	Nz,max=1.8
Starting point (ft)	431.1	513.6	542.2
Time elapsed (sec)	8.04	5.47	4.43
Overshoot (ft)	0	5	15

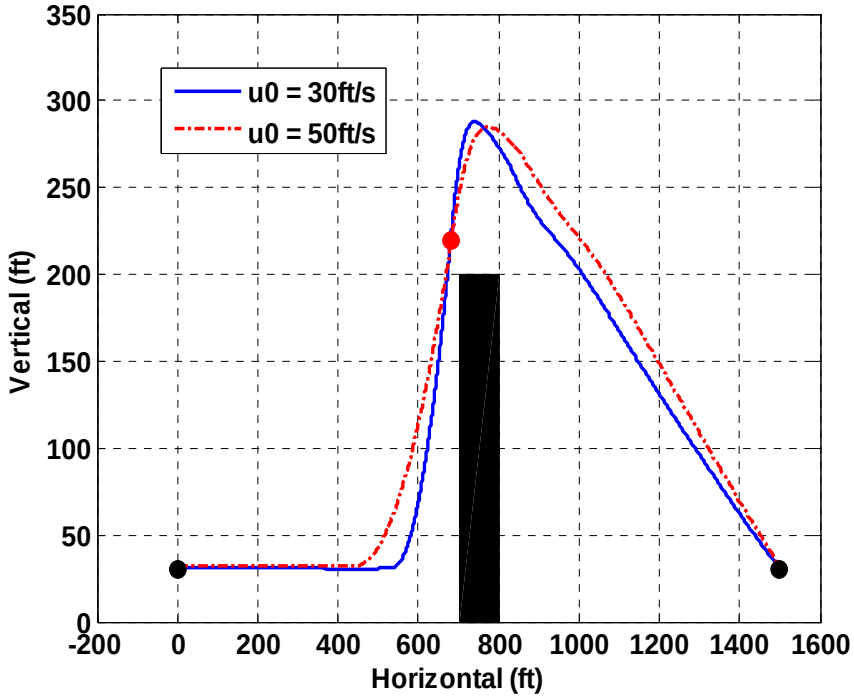


Figure 47. Trajectory response comparison for different initial speeds – neglected subsystems w/o vertical velocity constraint

Table 8. Simulation result comparison for different initial speeds – neglected subsystems w/o vertical velocity constraint

	u0=30 ft/s	u0=50 ft/s
Starting point (ft)	532.1	434.3
Time elapsed (sec)	4.99	4.93
Overshoot (ft)	70	70

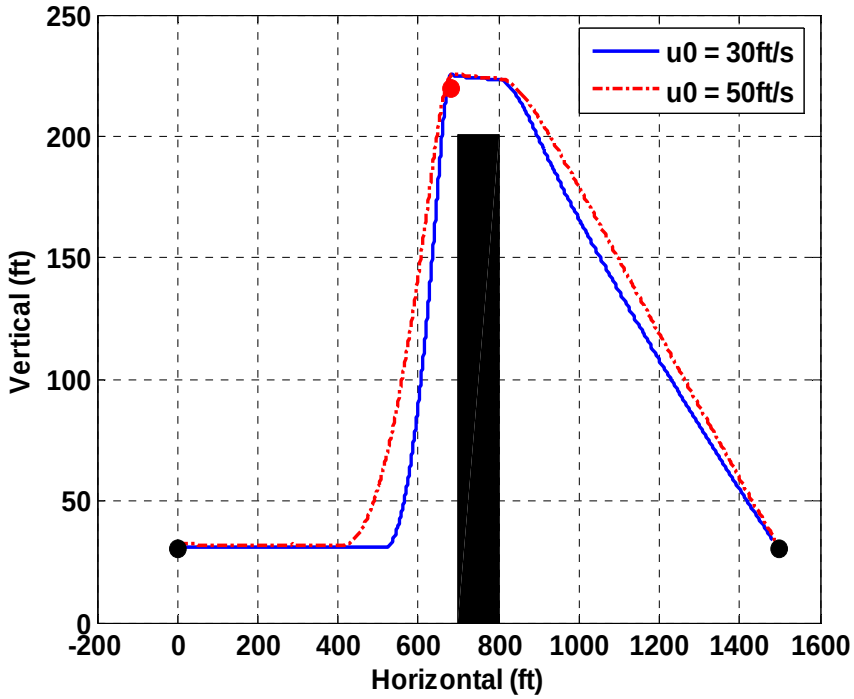


Figure 48. Trajectory response comparison for different initial speeds – neglected subsystems with vertical velocity constraint

Table 9. Simulation result comparison for different initial speeds – neglected subsystems with vertical velocity constraint

	u0=30 ft/s	u0=50 ft/s
Starting point (ft)	513.6	406.7
Time elapsed (sec)	5.47	5.45
Overshoot (ft)	5	5

Figures 49 through 55 show the GUST simulation results for the cases with the first-order representation of a UAV plus autopilot system dynamics. It is seen from Figure 49 that even though the obstacle is detected when the vehicle is at 400 *ft* away from it, the avoidance maneuver is not initiated by the algorithm until the vehicle reaches the point roughly 190 *ft* from the obstacle. Compared to the previous case shown in Figure 38, the vehicle needs to start the avoidance maneuver roughly 5 *ft* earlier in this case. It is seen from Figure 48 that, as per the terminal constraint, the vertical velocity becomes zero at the safe way point. Compared to the cases of neglected subsystem dynamics in Figures 39 and 40, it can be noticed from Figure 50 that the horizontal velocity remains close to the initial value until the vehicle reaches the safe waypoint. This means that the first-order representation of the vehicle plus autopilot system is good enough for a real-time optimization problem. Also, from Figure 51, it is seen that the load factor is maintained within its assumed limit of 1.5 during the avoidance maneuver. Body attitude angle responses and control input responses are shown in Figures 52 and 53, respectively.

The trajectory response comparisons for different load factor limits and for different initial speeds with the first-order approximation of the subsystem dynamics are shown in Figures 54 and 55, respectively. As seen in the previous cases, with a higher load factor limit, the vehicle can initiate obstacle avoidance maneuver later and maintain the original flight path longer. In addition, it is seen that with a faster initial speed, the vehicle should start obstacle avoidance maneuver earlier.

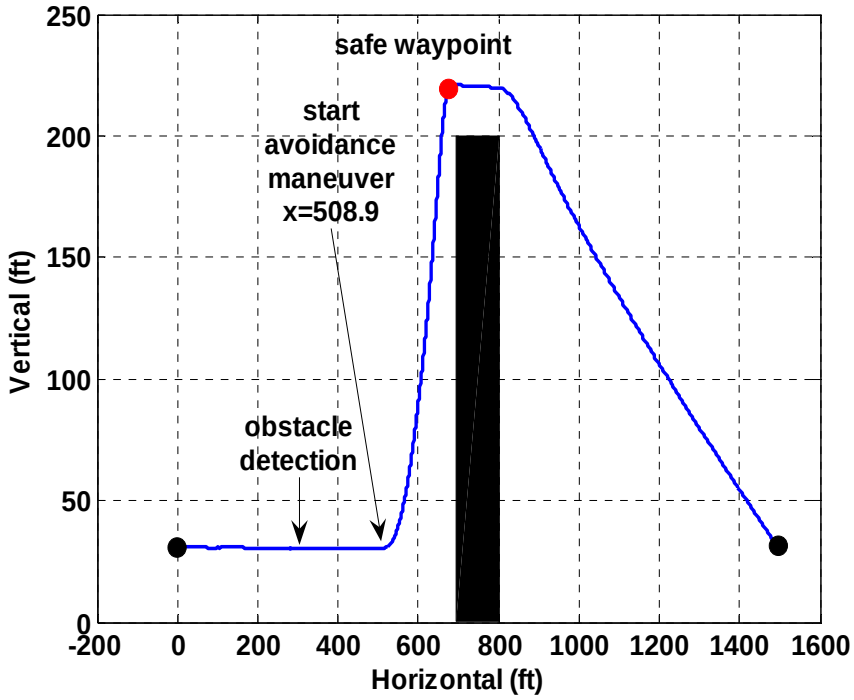


Figure 49. Trajectory response – 1st order model of subsystems with vertical velocity constraint

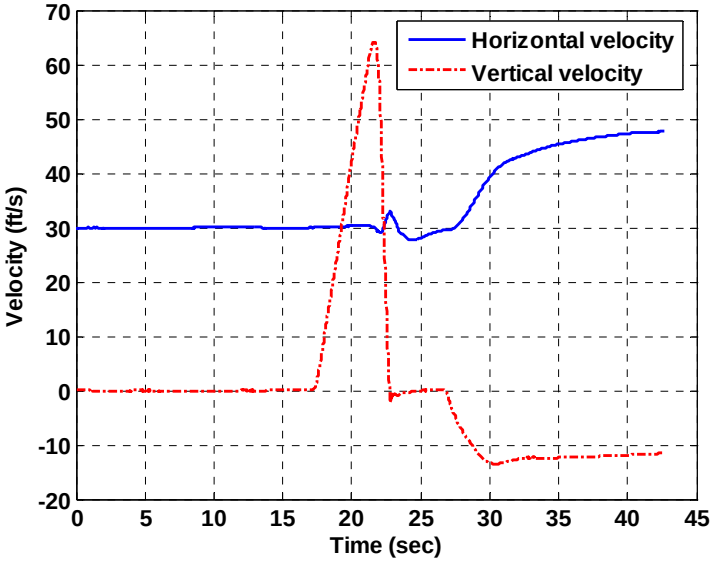


Figure 50. Velocity response – 1st order model of subsystems with vertical velocity constraint

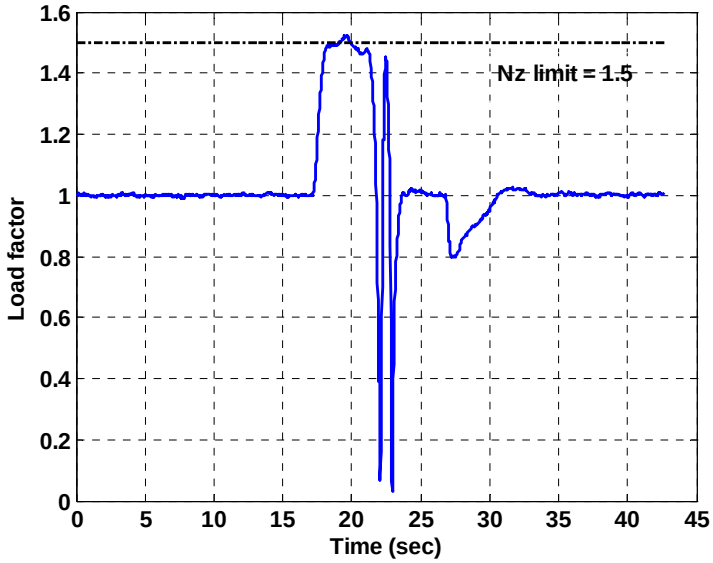


Figure 51. Load factor response – 1st order model of subsystems with vertical velocity constraint

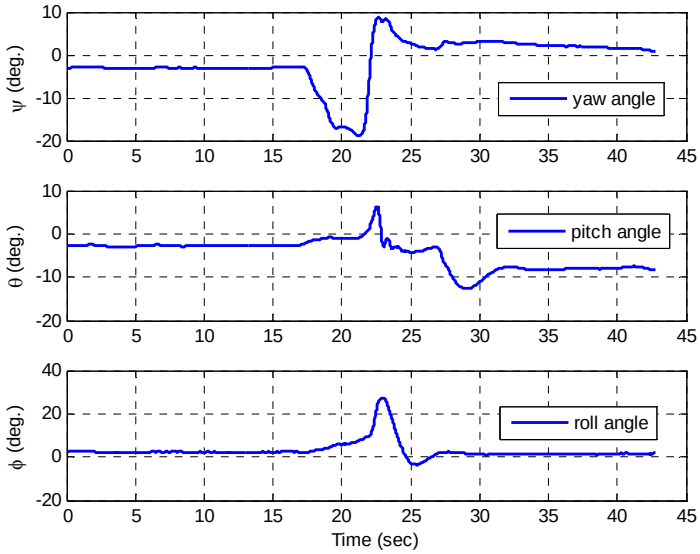


Figure 52. Body attitude responses – 1st order model of subsystems with vertical velocity constraint

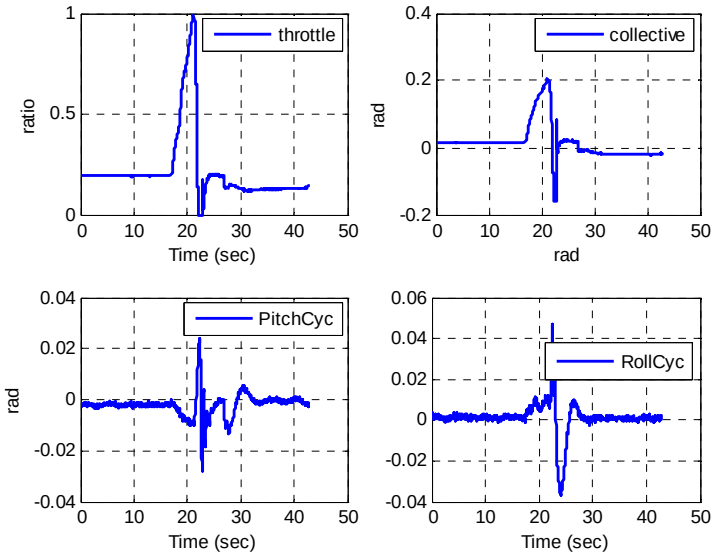


Figure 53. Control responses – 1st order model of subsystems with vertical velocity constraint

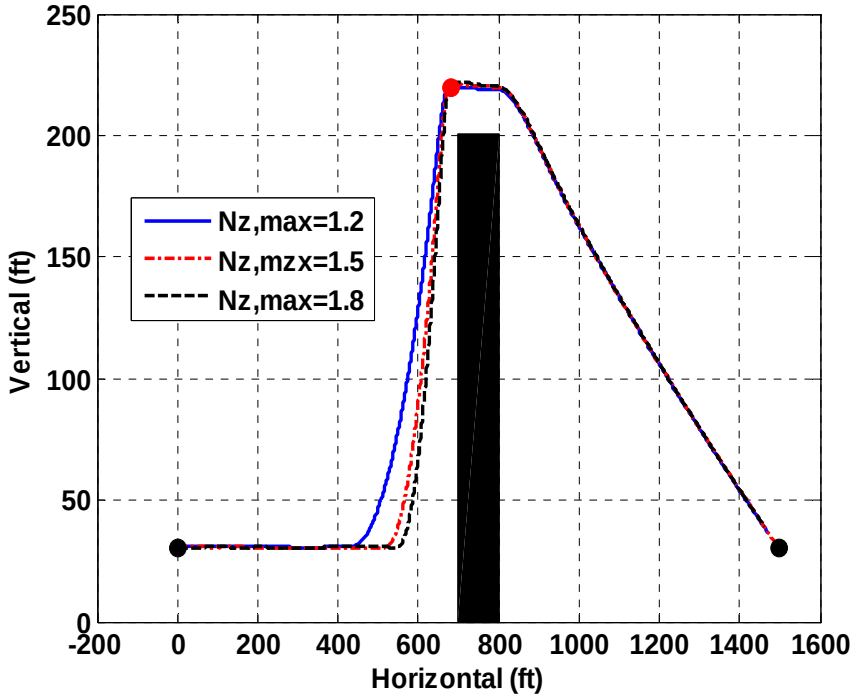


Figure 54. Trajectory response comparison for different load factor limits – 1st order model of subsystems with vertical velocity constraint

Table 10. Simulation result comparison for different load factor limits – 1st order model of subsystems with vertical velocity constraint

	Nz,max=1.2	Nz,max=1.5	Nz,max=1.8
Starting point (ft)	422.4	508.9	539.3
Time elapsed (sec)	8.58	5.71	4.66

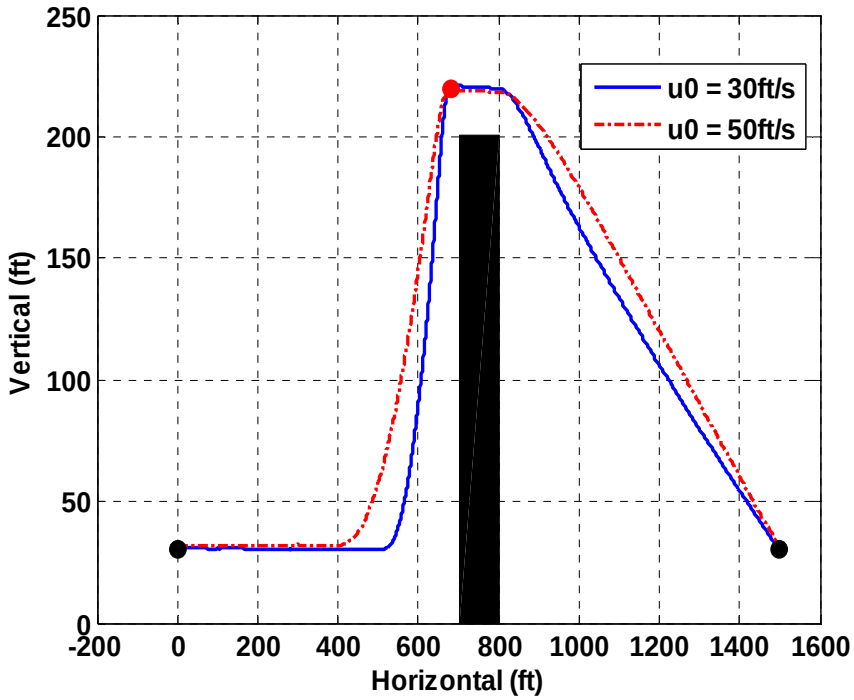


Figure 55. Trajectory response comparison for different initial speeds – 1st order model of subsystems with vertical velocity constraint

Table 11. Simulation result comparison for different initial speeds – 1st order model of subsystems with vertical velocity constraint

	u0=30 ft/s	u0=50 ft/s
Starting point (ft)	508.9	398.8
Time elapsed (sec)	5.71	5.68

Unsafe avoidance solution case

If an obstacle is detected when the vehicle is too close to the obstacle, it is possible that the solution from the NTG may not satisfy the requirement that the horizontal velocity must be non-negative during the obstacle avoidance maneuver. In order to study this case, the obstacle detection range is now set at 150 *ft* while keeping all other parameter values to be same as before (see Tables 4 and 5). Figure 56 shows the optimal trajectory from the NTG at the point when the obstacle is detected, i.e., when the vehicle is at 100 *ft* away from the obstacle. It is clear from the trajectory solution shown in Figure 56 that the requirement of non-zero horizontal velocity during the avoidance maneuver is not met by the NTG solution.

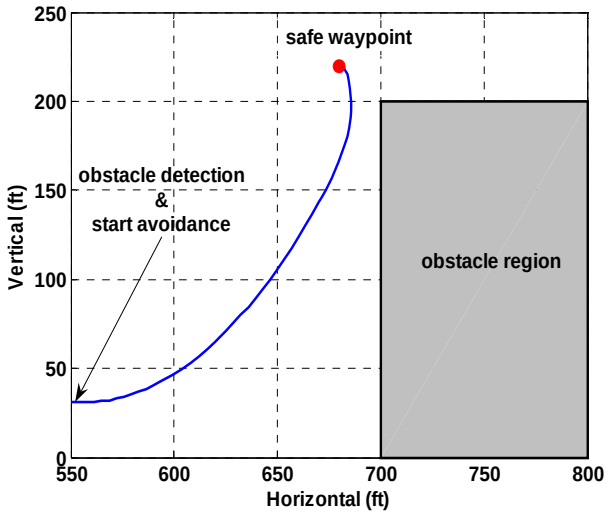


Figure 56. Unsafe avoidance trajectory solution.

The absence of a safe avoidance solution from the NTG results in initiation of a horizontal deceleration maneuver at a pre-selected value of -10 ft/s^2. Now, two alternatives exist for the decision on when to initiate an avoidance maneuver. The first alternative (strategy A) is to initiate an avoidance maneuver as soon as a safe avoidance solution (the solution satisfying the requirement that the horizontal velocity is non-negative during the avoidance maneuver) is obtained. Another alternative (strategy B) is

to decelerate the vehicle speed to the point when the optimal horizontal acceleration command from the NTG is zero (corresponding to the case of absolute minimum time avoidance solution). The simulation results for the two alternative strategies are shown in Figures 57 through 62.

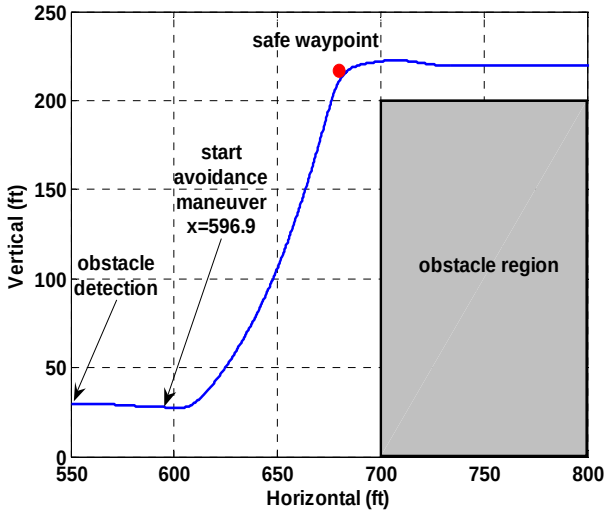


Figure 57. Safe obstacle avoidance trajectory response with strategy A for the case of obstacle detection too close to the obstacle.

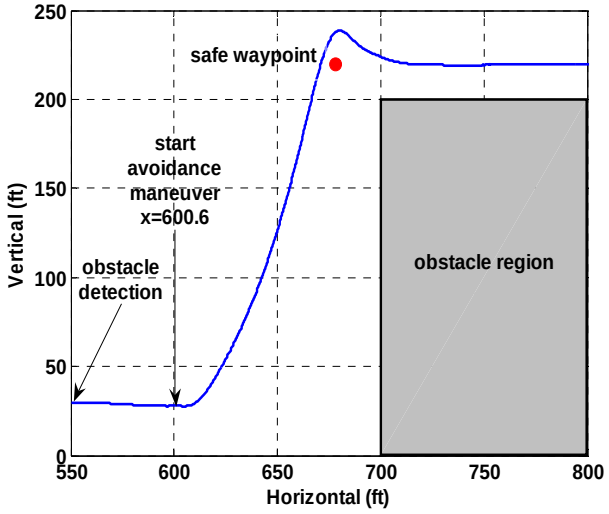


Figure 58. Safe obstacle avoidance trajectory response with strategy B for the case of obstacle detection too close to the obstacle.

From Figure 57 for strategy A, it is seen that when the vehicle arrives at the point (597, 0, 30), a safe avoidance solution becomes feasible, and the avoidance maneuver is initiated at this point. From Figure 58 for strategy B, the horizontal deceleration of the vehicle continues till the point (601, 0, 30) at which the optimal horizontal acceleration command from the NTG becomes zero, and the avoidance maneuver is initiated at this point. Figures 59 and 60 show a comparison of the horizontal and vertical velocity changes for strategies A and B, respectively. It is seen from Figures 59 and 60 that the vehicle decelerates to roughly 18 *ft/s* prior to the avoidance maneuver for strategy A whereas the vehicle decelerates to roughly 15 *ft/s* before the avoidance maneuver is initiated for strategy B. The load factor responses shown in Figures 61 and 62 for the two strategies indicate that the load factor stays within the assumed limit of 1.5 as required in both cases. Also, control responses are shown in Figures 63 and 64 for the two strategies.

It is seen that the time elapsed from the point of obstacle detection to reach the safe waypoint for strategy A is roughly 6 seconds whereas the same for strategy B is close to 7 seconds. However, for strategy B, the vehicle continues along the original flight path a litter longer (the avoidance maneuver is initiated roughly 103 *ft* away from the obstacle for strategy B in contrast to roughly 99 *ft* for strategy A). These results indicate that for the case of obstacle detection too close to the obstacle resulting in an unsafe avoidance solution, the vehicle must decelerate first before an avoidance maneuver is initiated. The decision on the extent of deceleration (speed change) actually used is very much dependent on a tradeoff between the elapsed distance along the original flight path versus the elapsed time for reaching the safe way point.

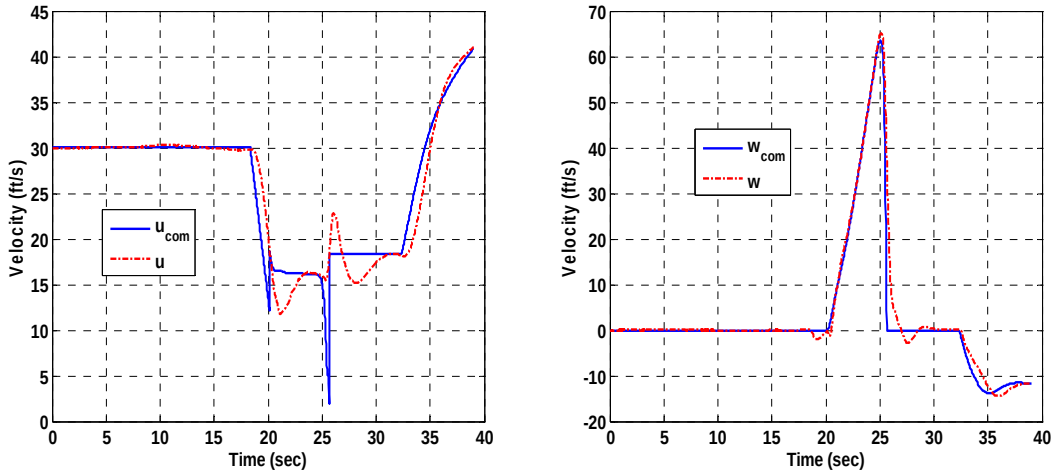


Figure 59. Horizontal and vertical components of velocity response with strategy A for the case of obstacle detection too close to the obstacle.

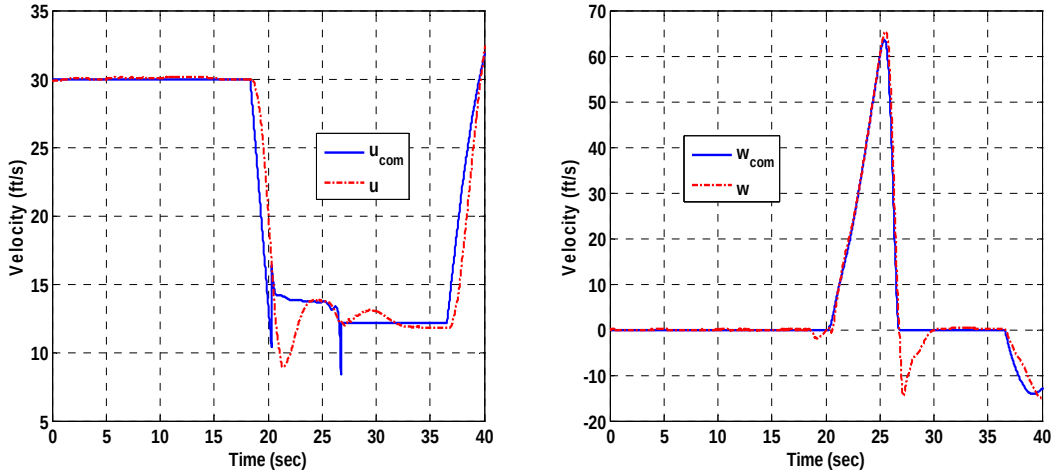


Figure 60. Horizontal and vertical components of velocity response with strategy B for the case of obstacle detection too close to the obstacle.

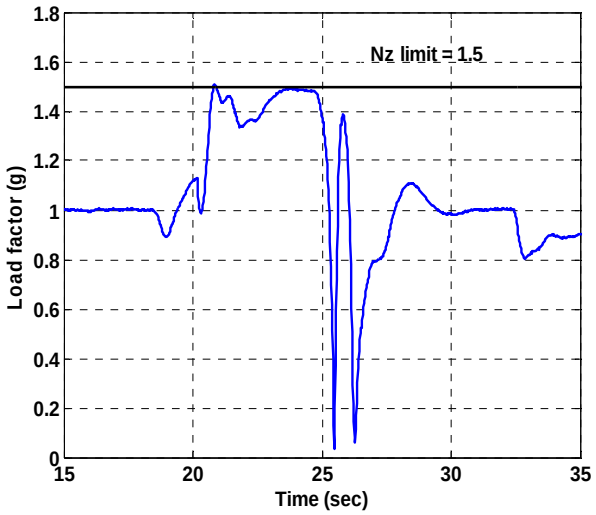


Figure 61. Load factor response with strategy A for the case of obstacle detection too close to the obstacle.

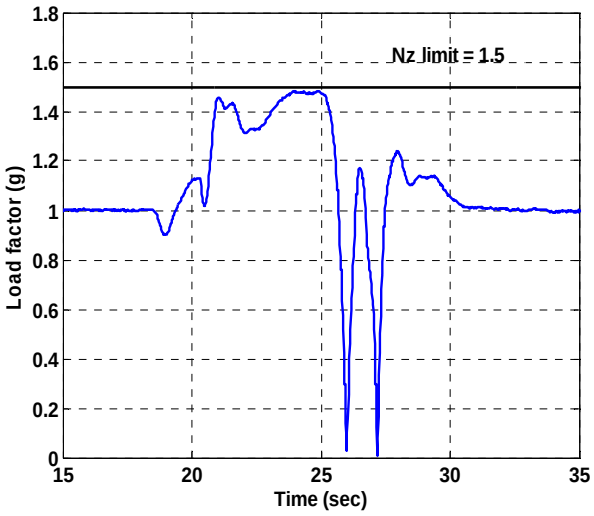


Figure 62. Load factor response with strategy B for the case of obstacle detection too close to the obstacle.

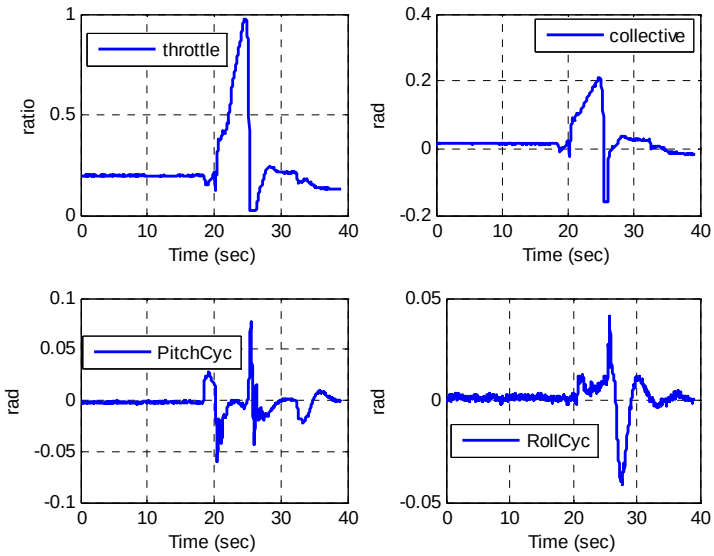


Figure 63. Control responses with strategy A for the case of obstacle detection too close to the obstacle.

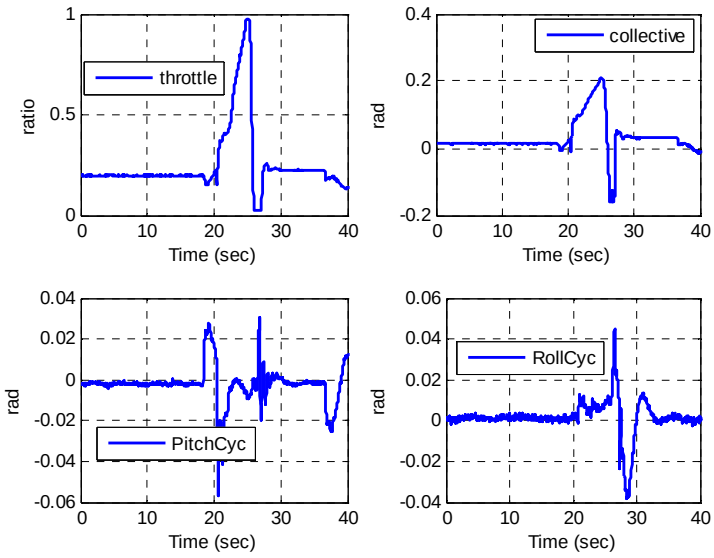


Figure 64. Control responses with strategy B for the case of obstacle detection too close to the obstacle.

Multiple safe waypoints case

In obstacle avoidance, there may exist several maneuvers for a vehicle. For example, one may consider a pull-up maneuver or a turning maneuver or a combination for avoiding a building, while staying within the flight envelope of the vehicle. An approach for avoidance maneuver selection in case of multiple safe waypoints is described in the previous section. The proposed approach is evaluated in simulations by considering avoidance of three different size obstacles while constrained by the load factor limit.

Obstacle 1: width x height x depth (340x200x100 ft)

Obstacle 2: width x height x depth (560x200x100 ft)

Obstacle 3: width x height x depth (150x200x100 ft) and (100x100x100 ft)

The obstacle detection range is assumed to be 400 *ft*. The parameters of the B-spline representations of flat outputs within NTG are extended to the lateral variable *y* as given in Table 12.

Table 12. Details of flat outputs within NTG for obstacle avoidance with load factor limiting and with multiple safe waypoints.

Flat outputs	intervals	order	multiplicity	# of coeffi.
x	5	6	4	14
y	5	6	4	14
z	5	6	4	14
N_z	5	2	1	6
t_f	1	1	0	1

Obstacle 1:

For obstacle 1 with the selected dimensions of 340 *ft* width, 200 *ft* height and 100 *ft* depth, the safe waypoint 1 in the vertical plane is set at (680, 0, 220) and the safe waypoint 2 in the horizontal plane is set at (680, 190, 30). Both are chosen based on 20 *ft* clearance distance from the obstacle boundary. The proposed algorithm solves the time-optimal control problem for each safe waypoint and chooses the better of the two solutions that will result in a greater extent of staying along the original flight.

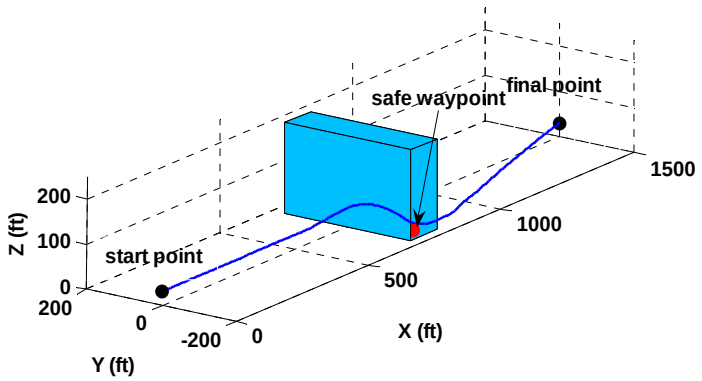


Figure 65. Avoidance trajectory with obstacle 1

At the instance of obstacle detection (400 *ft* from the obstacle), the required horizontal acceleration commands from the time-optimal solutions are positive for both safe waypoints, and the vehicle maintains the original flight. When the vehicle is at **x=508.9 *ft***, the required horizontal acceleration command becomes 0 in order to reach the safe waypoint 1. However, the required horizontal acceleration command to reach the safe waypoint 2 is still positive at that instance. Hence, the vehicle continues to maintain its original flight, and the safe waypoint 2 is selected by the algorithm for obstacle avoidance. When it reaches **x=531.7 *ft***, the required horizontal acceleration command

becomes zero and a turning maneuver is initiated toward the safe waypoint 2. The resulting inertial axes components of the vehicle velocity response and the velocity commands are compared in Figures 66 and 67. It can be noticed that the vertical velocity is induced to initiate a rapid turning motion. The load factor response for avoidance of the selected obstacle 1 is shown in Figure 68.

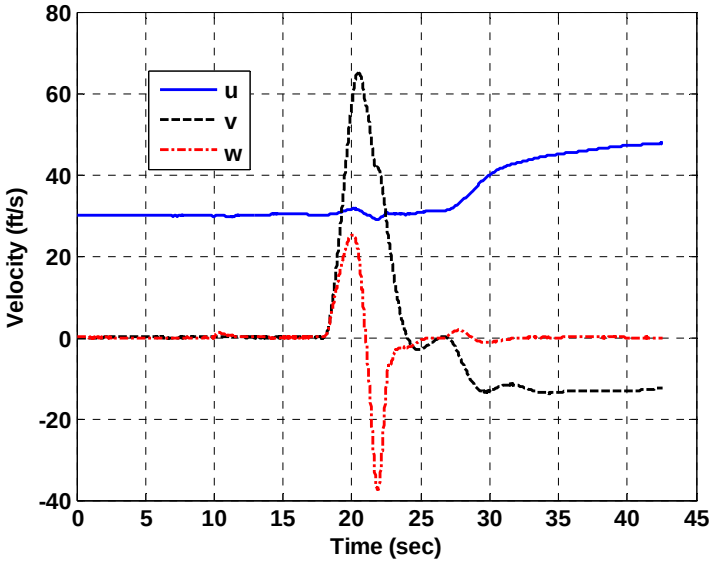


Figure 66. Horizontal, lateral and vertical components of velocity response for the case of avoidance of obstacle 1 while constrained by the load factor limit.

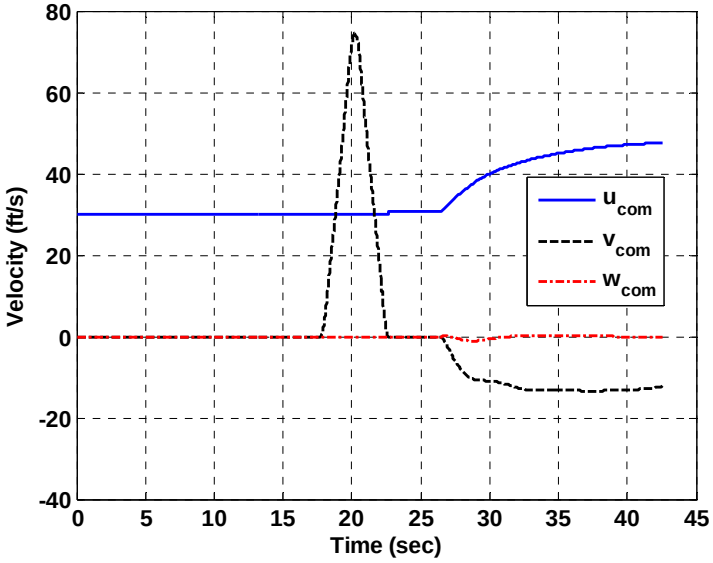


Figure 67. Horizontal, lateral and vertical components of velocity command for the case of avoidance of obstacle 1 while constrained by the load factor limit.

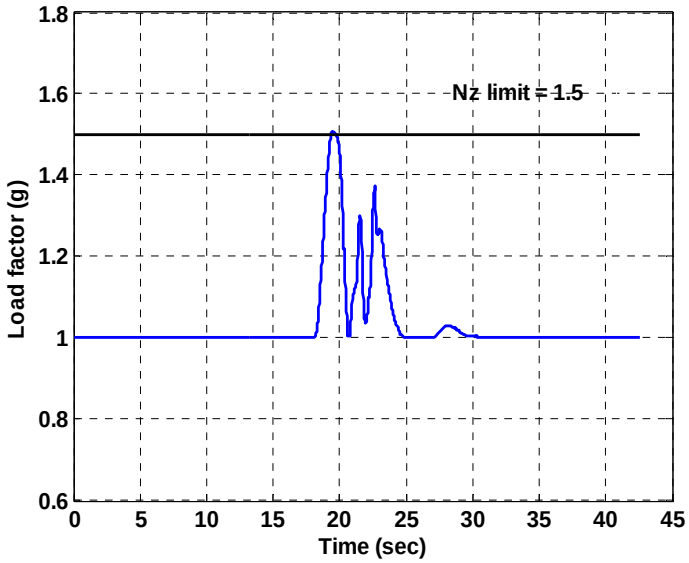


Figure 68. Load factor response for the case of avoidance of obstacle 1 while constrained by the load factor limit.

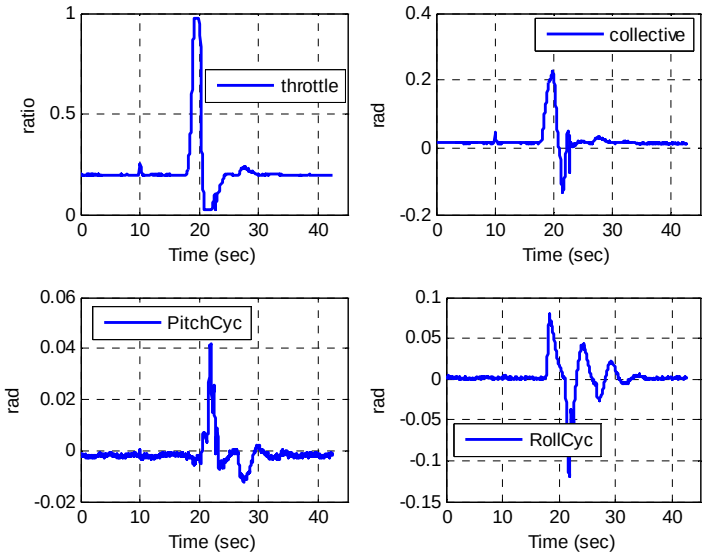


Figure 69. Control responses for the case of avoidance of obstacle 1 while constrained by the load factor limit.

Obstacle 2:

For obstacle 2 with the selected dimensions of 560 *ft* width, 200 *ft* height and 100 *ft* depth, the safe waypoint 1 in the vertical plane is set to be the same as that for the

obstacle 1 (the selected heights of both obstacles are the same) case, and the safe waypoint 2 in the horizontal plane is set at (680, 300, 30). At the instance of obstacle detection (400 *ft* from obstacle), the required horizontal acceleration commands from the time-optimal solutions to the two selected safe waypoints are positive, and the vehicle maintains the original flight. When the vehicle arrives at the point (**491.8, 0, 30**), the required horizontal acceleration command to reach the safe waypoint 2 in minimum time becomes 0. However, the required horizontal acceleration command to reach the safe waypoint 1 in minimum time is still positive. Hence, the vehicle maintains its original flight path, and the waypoint 1 is selected by the algorithm for obstacle avoidance. When it arrives at the point (**508.9, 0, 30**), the required horizontal acceleration command for the safe waypoint 1 becomes zero, and the vehicle executes a pull-up maneuver toward the safe waypoint 1. The resulting trajectory response, the inertial axes components of velocity and the load factor response for avoidance of the selected obstacle 2 of 560 *ft* width and 200 *ft* height are shown in Figures 70, 71 and 72, respectively.

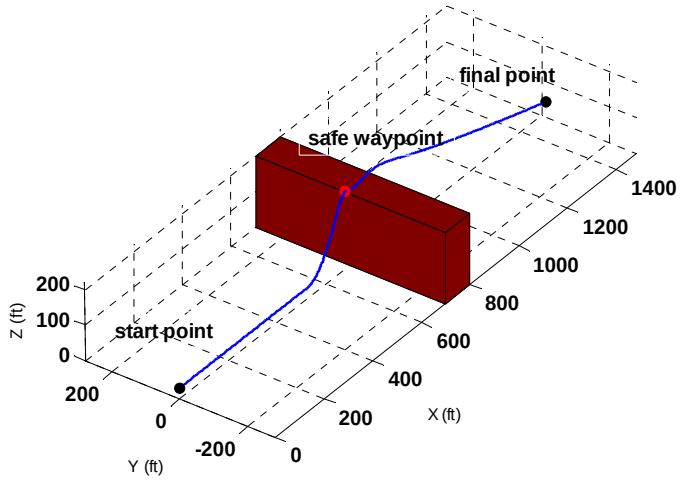


Figure 70. Avoidance trajectory with obstacle 2

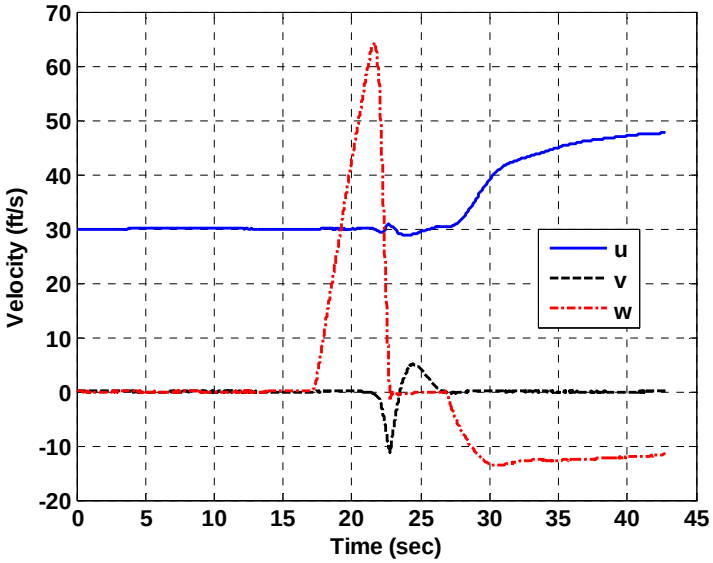


Figure 71. Horizontal, lateral and vertical components of velocity response for the case of avoidance of obstacle 2 while constrained by the load factor limit.

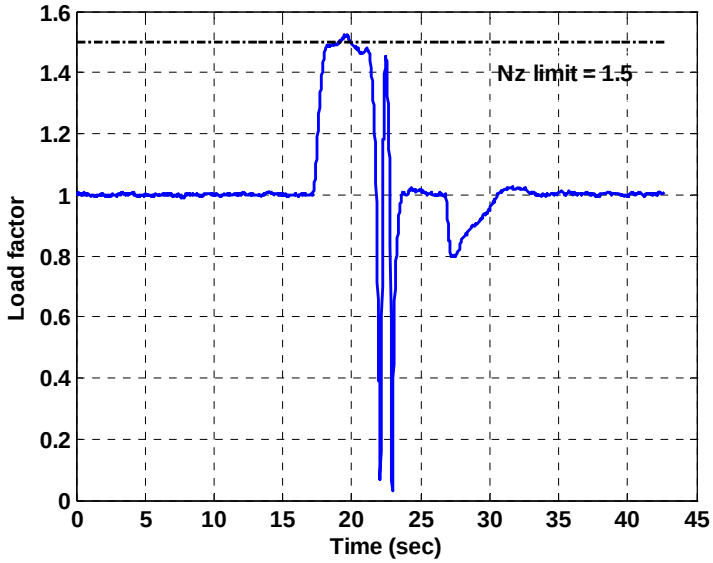


Figure 72. Load factor response for the case of avoidance of obstacle 2 while constrained by the load factor limit.

Obstacle 3:

In this case, it is assumed that one obstacle with the selected dimensions of 150 *ft* width, 200 *ft* height and 100 *ft* depth is next to another obstacle with the selected dimensions of 100 *ft* width, 100 *ft* height and 100 *ft* depth. The safe waypoint 1 in the vertical plane is set at (680, 0, 220) and the safe waypoint 2 is set at (680, 50, 120). the

required horizontal acceleration commands from the time-optimal solutions are positive for both safe waypoints, and the vehicle maintains the original flight. When the vehicle is at **x=508.9 *ft***, the required horizontal acceleration command becomes 0 in order to reach the safe waypoint 1. However, the required horizontal acceleration command to reach the safe waypoint 2 is still positive at that instance. Hence, the vehicle continues to maintain its original flight, and the safe waypoint 2 is selected by the algorithm for obstacle avoidance. When it reaches **x=556.0 *ft***, the required horizontal acceleration command becomes zero and avoidance maneuver is initiated toward the safe waypoint 2. This maneuver looks like a combination of the turning and pull up maneuver. The trajectory response is shown in Figure 73. Also, the resulting inertial axes components of the vehicle velocity response and the velocity command are compared in Figures 74 and 75. The load factor response for avoidance of the selected obstacle 3 is shown in Figure 76.

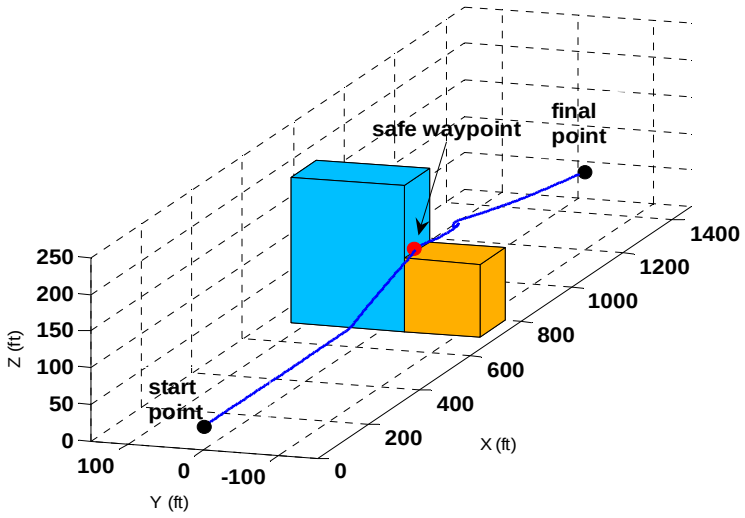


Figure 73. Avoidance trajectory with obstacle 3

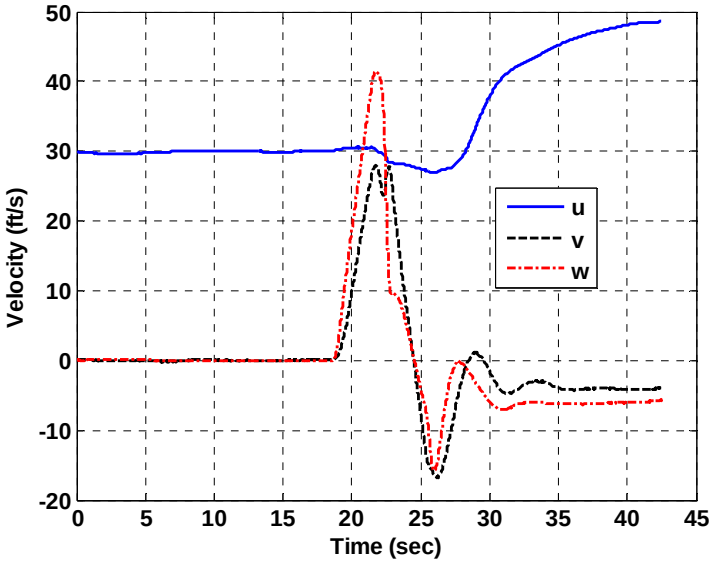


Figure 74. Horizontal, lateral and vertical components of velocity response for the case of avoidance of obstacle 3 while constrained by the load factor limit.

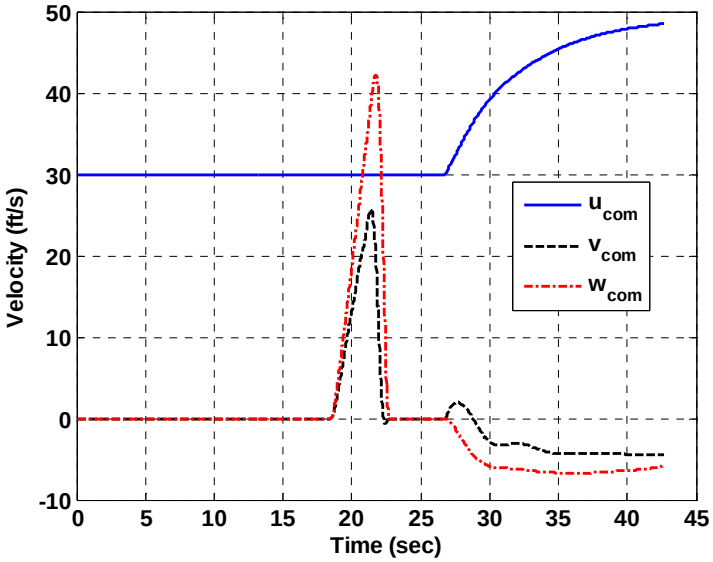


Figure 75. Horizontal, lateral and vertical components of velocity command for the case of avoidance of obstacle 3 while constrained by the load factor limit.

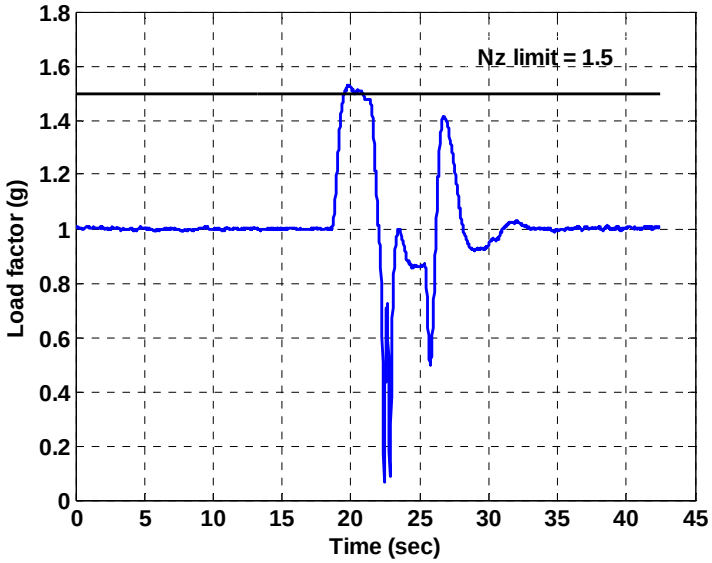


Figure 76. Load factor response for the case of avoidance of obstacle 3 while constrained by the load factor limit.

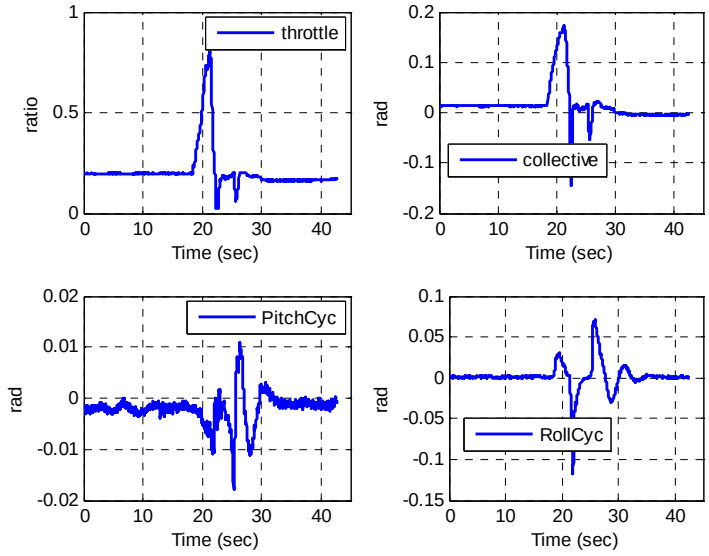


Figure 77. Control responses for the case of avoidance of obstacle 3 while constrained by the load factor limit.

Unidentified obstacle size case

During the obstacle detection, it may happen that the obstacle detection device sees only a part of an obstacle due to its own performance limit. Then, the proposed

algorithm determines safe obstacle avoidance maneuvers based on the insufficient information while maintaining the limit parameter within the boundary. In this example, instead of using a detailed model of the obstacle detection device, it is assumed that information on the obstacle size is updated every second. Obstacle avoidance maneuvers in the vertical plane are considered in this case, and an obstacle with the selected dimensions of 200 *ft* height and 100 *ft* depth is considered in this case.

It is assumed that the height of the building is identified as 150 *ft* when it is detected, and the first safe waypoint is set at (680, 0, 170). For this safe waypoint, the horizontal acceleration command becomes zero at **X=533.7** *ft*, and the vehicle starts the obstacle avoidance maneuver. After the initiation of the obstacle avoidance maneuver, it is assumed that the obstacle is found to be higher than at previous instance of the detection, and new safe waypoints change to (680, 0, 200) and (680, 0, 220) every one second later. Figure 78 compares trajectory responses for two cases: the case of fully identified obstacle size at the moment of detection (red dashed line), and the case of not fully identified obstacle size (blue line).

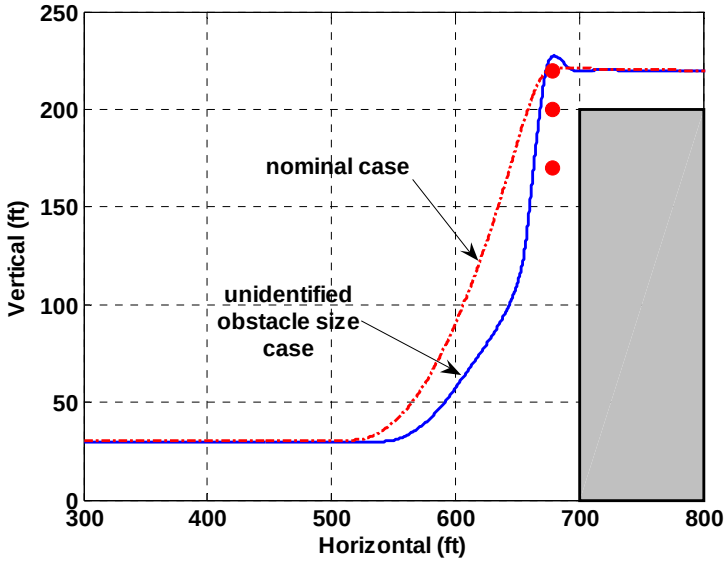


Figure 78. Obstacle avoidance trajectory for the case of unidentified obstacle size.

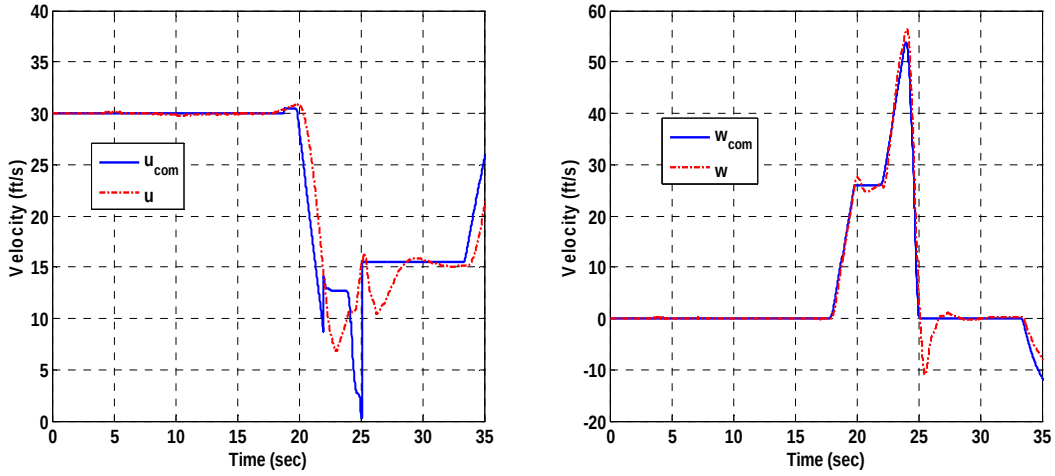


Figure 79. Velocity command versus velocity response for the case of unidentified obstacle size.

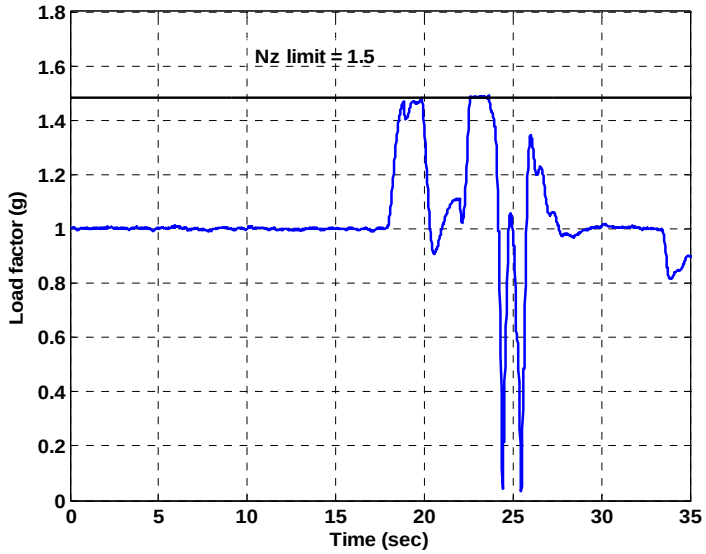
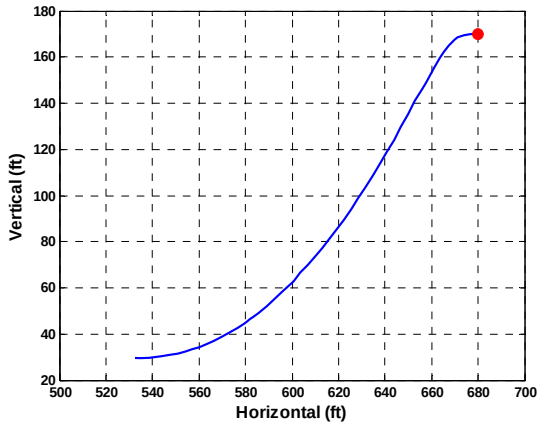


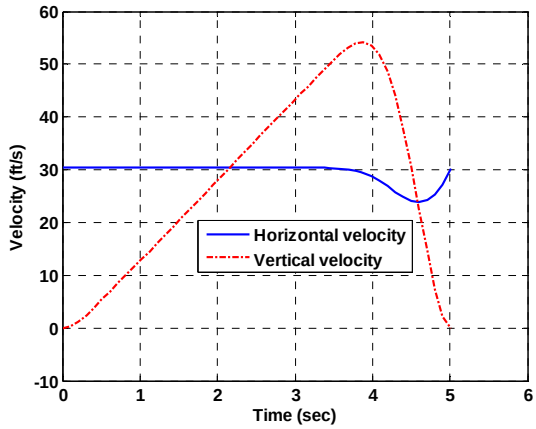
Figure 80. Load factor response for the case of unidentified obstacle size.

Figures 79 and 80 show the velocity response and load factor response during the obstacle avoidance maneuver, respectively. Figures 81 through 84 show optimal solutions from the NTG at different instances. Figure 81 presents optimized trajectory, velocity, and acceleration from the NTG when the avoidance maneuver is initiated for the point

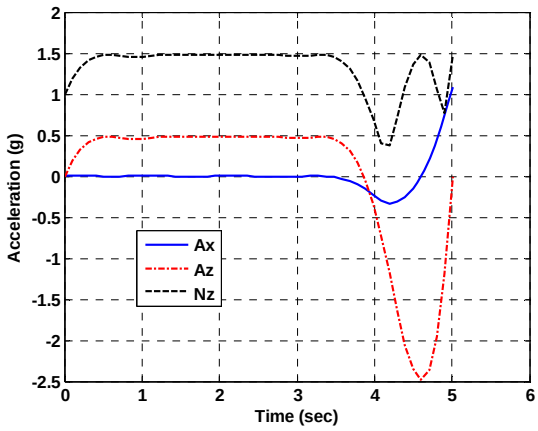
(680, 0, 170). New safe waypoint is set to the point (680, 0, 200) at 1 second after initiating the avoidance maneuver, and the optimal solution from the NTG at that instance is shown in Figure 82. Another 1 second later, a final safe waypoint is set to the point (680, 0, 220). The NTG solution at that moment is provided in Figure 83. As shown in Figure 83, the optimal solution at this time is unsafe avoidance solution since the horizontal velocity should be negative. Thus, a horizontal deceleration command is applied while a vertical acceleration is commanded to zero. The velocity response between 20 seconds and 23 seconds in Figure 79 can be explained by the unsafe avoidance solution. After horizontal deceleration during roughly 3 seconds, a safe avoidance solution is obtained from the NTG, and the optimized results are shown in Figure 84. Time elapsed for the obstacle avoidance maneuver is 6.91 seconds, and it takes much longer to reach the final safe waypoint when compared to the nominal case (5.71 seconds). As shown in Figure 80, maximum load factor is used during the obstacle avoidance maneuver. Figures 85 and 86 show body attitude responses and control responses, respectively.



(a) Trajectory solution from NTG

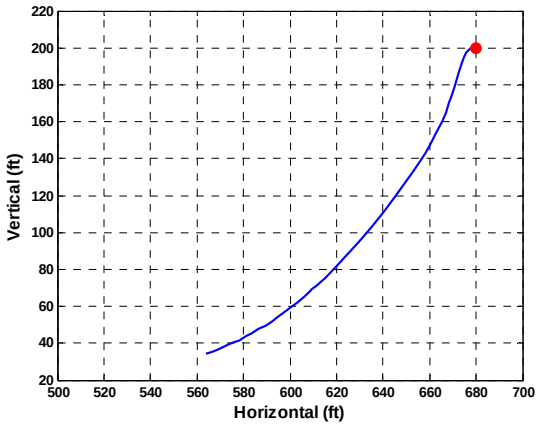


(b) Velocity solution form NTG

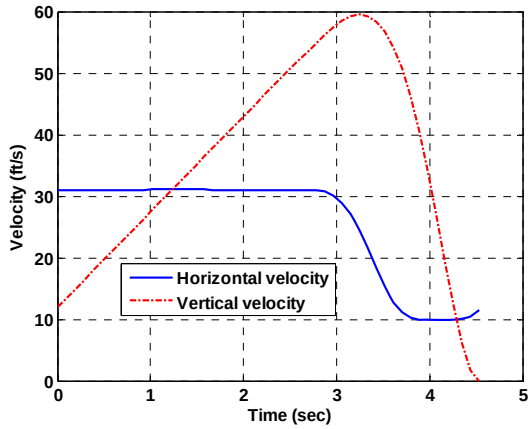


(c) Acceleration solution from NTG

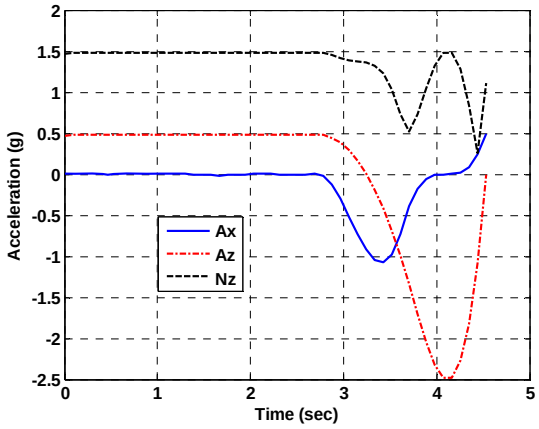
Figure 81. NTG solution at the initiation of avoidance maneuver.



(a) Trajectory solution from NTG

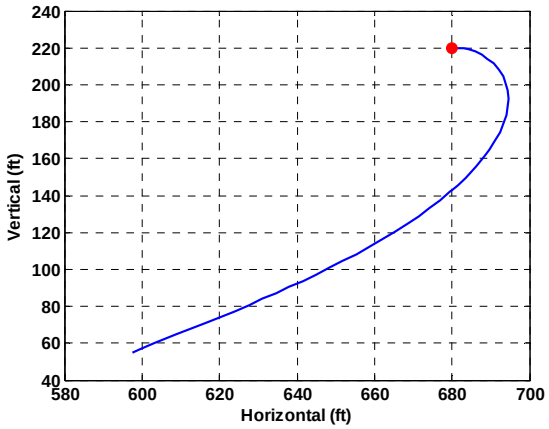


(b) Velocity solution form NTG

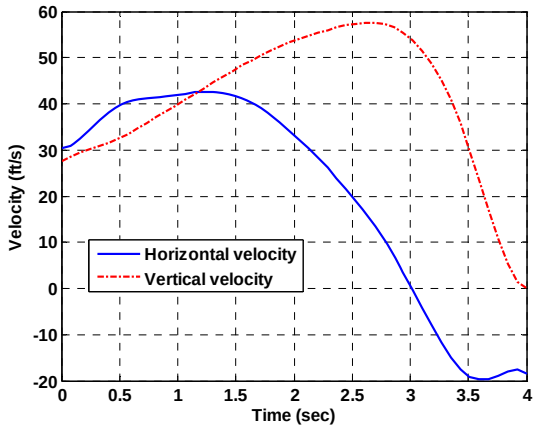


(c) Acceleration solution from NTG

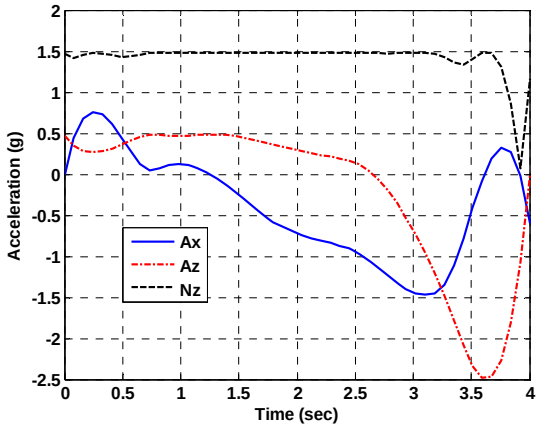
Figure 82. NTG solution at 1 second after the initiation of avoidance maneuver.



(a) Trajectory solution from NTG

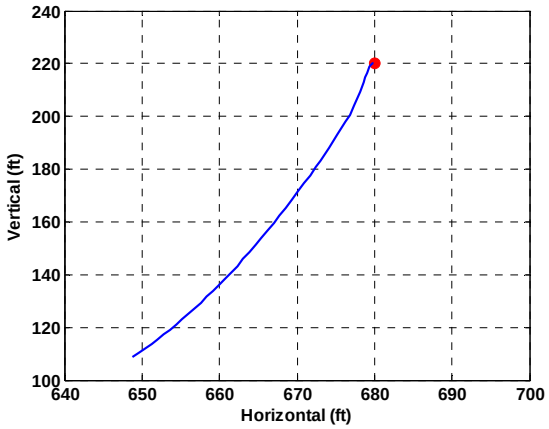


(b) Velocity solution form NTG

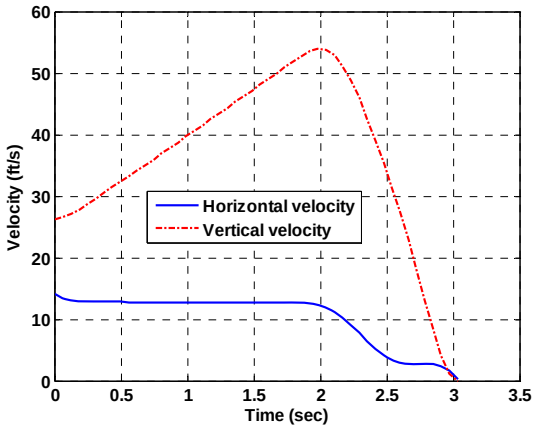


(c) Acceleration solution from NTG

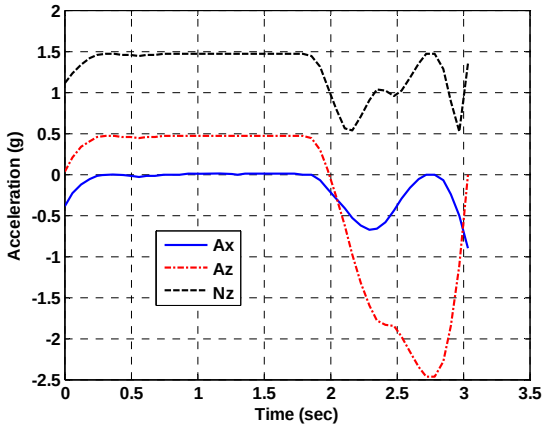
Figure 83. NTG solution at 2 seconds after the initiation of avoidance maneuver.



(a) Trajectory solution from NTG



(b) Velocity solution form NTG



(c) Acceleration solution from NTG

Figure 84. NTG solution at 5 seconds after the initiation of avoidance maneuver.

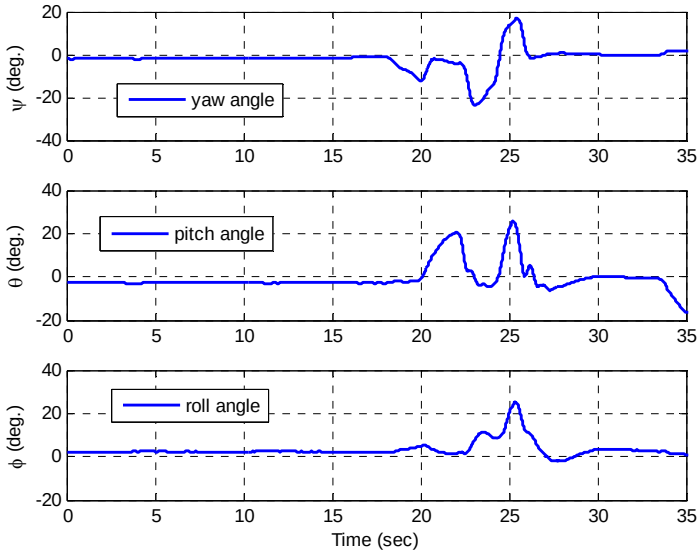


Figure 85. Body attitude responses for the case of unidentified obstacle size.

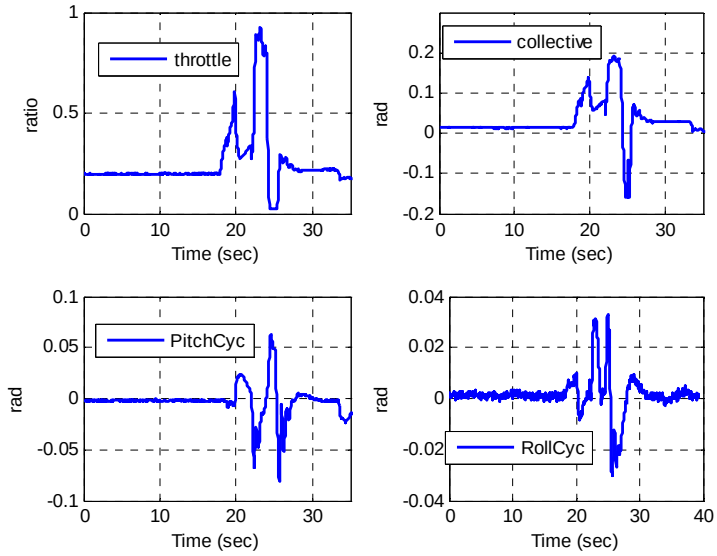


Figure 86. Control responses for the case of unidentified obstacle size.

4.5.2 Longitudinal flapping angle limiting

Aggressive maneuvers for obstacle avoidance may result in excessive flapping deflection which may give rise to mast bumping in a teetering rotor system. In order to

evaluate the proposed algorithm for the case of rotor flapping limits, the obstacle avoidance problem in the vertical plane constrained by longitudinal flapping limits is considered. The longitudinal flapping dynamics is approximated by [5]

$$\dot{\beta}_{lon} = -\frac{1}{\tau}\beta_{lon} - q \tag{92}$$

where τ is $16/\gamma\Omega$. In the expression for τ, the parameter γ is the non-dimensional Lock's inertia number of the rotor blade and Ω is the rotor rotational speed in rad/sec. It is to be noted that, while the above simplified model for the longitudinal flapping dynamics is used in the NTG solutions, the nonlinear simulation model within the GUST makes use of a more comprehensive coupled longitudinal and lateral flapping dynamics model with several additional terms. In order to convert equation 92 into a differential flatness form as required for NTG, the pitch rate variable in equation 92 is replaced by an approximate equation in terms of the ratio of the vertical acceleration of the vehicle and its speed. With this simplification, equation 92 becomes

$$\dot{\beta}_{lon} \approx -\frac{1}{\tau}\beta_{lon} - \frac{a_z}{\sqrt{u^2 + w^2}} \tag{93}$$

From equations 58 and 93, it is easy to see that the relative degree of the assumed simplified longitudinal flapping dynamics model is 2.

The obstacle avoidance in the vertical plane with the load factor limit considered previously is simulated now with the longitudinal flap limit instead of the load factor limit. Table 13 gives the details of the simulation parameters used. Two values of flapping limit are considered: $\pm 3^0$ and $\pm 5^0$. The details of the B-spline representations within NTG are given in Table 14. The time constant τ in equation 93 is set at 0.055 sec based on the GTMax data.

Figures 87 and 88 show trajectory results for the two cases of assumed $\pm 3^0$ and $\pm 5^0$ of flapping limits, respectively. A zoom-in comparison of the results in Figures 87 and 88 is shown as Figure 89. It is seen from Figure 88 that for the case of $\pm 5^0$ flapping limit, the vehicle maintains the original flight mode longer (roughly 35 *ft* in this example) than for the case of $\pm 3^0$ flapping limit. This result is as expected since with a larger flapping limit, the vehicle can use larger acceleration commands while executing the avoidance maneuver, hence, the vehicle can maintain the original flight path a little longer prior to initiating an avoidance maneuver.

Table 13. Flight parameters for obstacle avoidance with flapping limits.

Initial position of UAV (*ft*)	(0, 0, 30)
Destination waypoint (*ft*)	(1500, 0,30)
Initial Speed (*ft/s*)	30
Flapping angle limit	$\pm 3^o / \pm 5^o$
Maximum detection range (*ft*)	400
Field of view	60˚

Table 14. Details of flat outputs within NTG for obstacle avoidance with flapping limits.

Flat outputs	intervals	order	multiplicity	# of coeffi.
x	5	6	4	14
z	5	6	4	14
β_{lon}	5	3	2	7
t_f	1	1	0	1

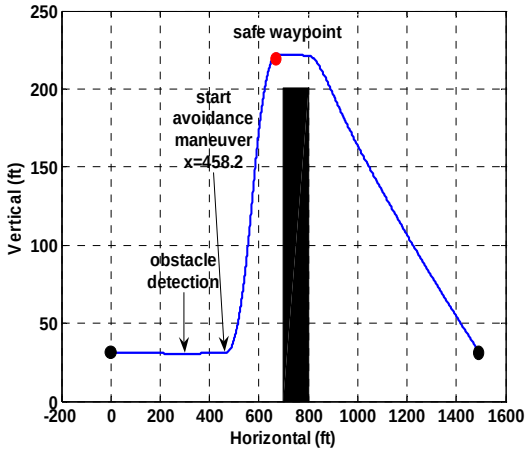


Figure 87. Obstacle avoidance trajectory with $\pm 3^0$ of flapping limit.

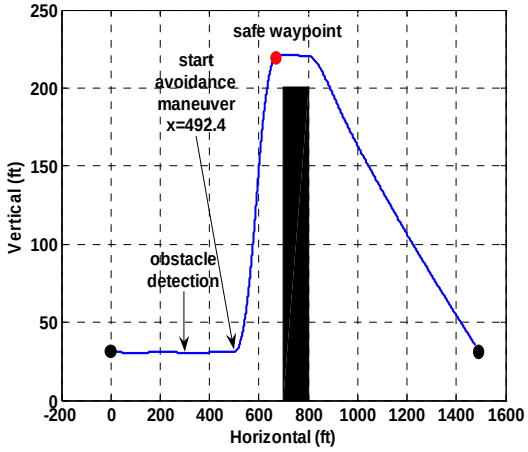


Figure 88. Obstacle avoidance trajectory with $\pm 5^0$ of flapping limit.

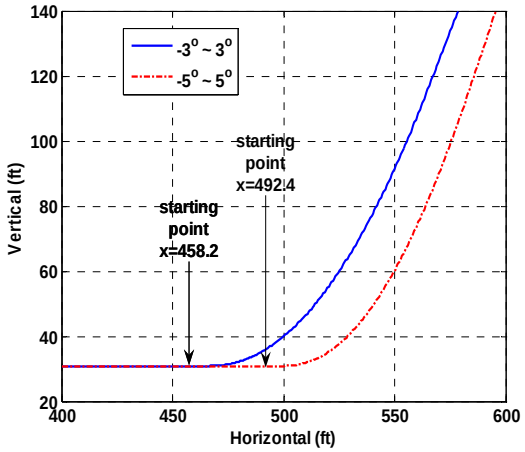


Figure 89. A zoom-in comparison of obstacle avoidance trajectories.

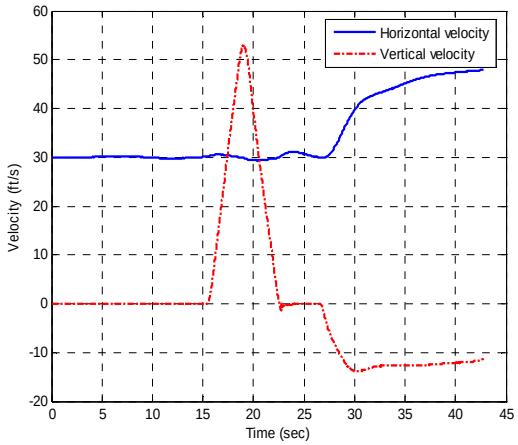


Figure 90. Horizontal and vertical components of velocity response for the case of obstacle avoidance with $\pm 3^0$ of flapping limits.

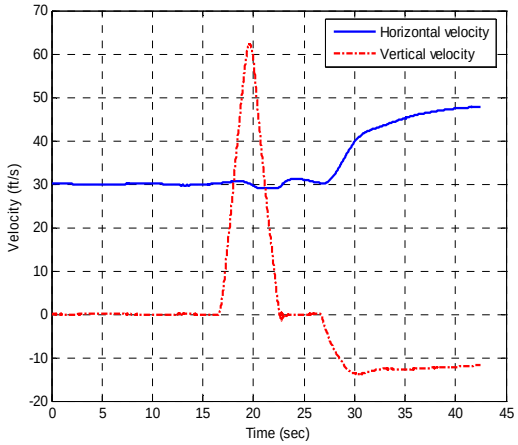


Figure 91. Horizontal and vertical components of velocity response for the case of obstacle avoidance with $\pm 5^0$ of flapping limits.

Figures 90 and 91 show horizontal and vertical components of velocity responses for the two cases of flapping limits. In both cases, the vertical velocity becomes zero at the safe way point as required. The elapsed time during the avoidance maneuver for the 5^o flapping limit is 6.9 sec and the same for the 3^o flapping limit case is 8.0 sec. This result is as expected considering that with a larger flapping limit, the vehicle can execute a more aggressive maneuver.

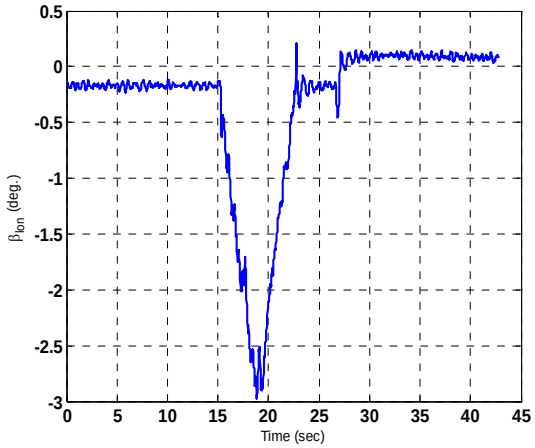


Figure 92. Longitudinal flapping response for the case of obstacle avoidance with $\pm 3^0$ of flapping limits.

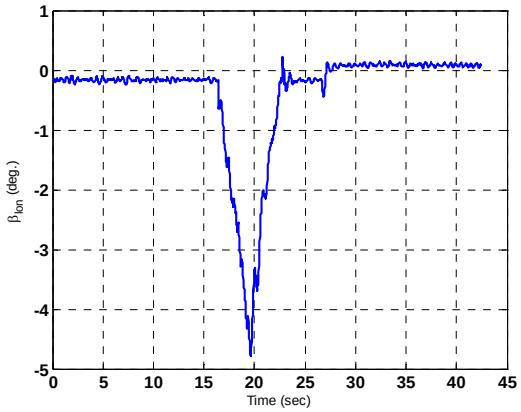


Figure 93. Longitudinal flapping response for the case of obstacle avoidance with $\pm 5^0$ of flapping limits.

The longitudinal flapping responses for the two cases of flapping limits considered are shown in Figures 92 and 93. In both cases, it is seen that the longitudinal flapping responses stay within the assumed flapping limits. It is interesting that one can compare the resulting aggressiveness of the avoidance maneuver between the load factor limit case and the flapping limit case by comparing the resulting load factor responses for the two cases. Figures 94 and 95 show the load factor responses with $\pm 3^0$ and $\pm 5^0$ flapping limits, respectively. It is noticed that the maximum load factor response for the $\pm 3^0$ flapping limit case is less than 1.4 and the same for the $\pm 5^0$ case is less than 1.5.

Since these represent lower aggressiveness when one compares to the maximum load factor response of 1.5 for the load limit case, as expected, the avoidance maneuvers are initiated earlier for the two cases of flapping limits considered in this study when compared to the load factor limiting case. Figure 96 and 97 show body attitude responses and control responses for the case of $\pm 3^0$ of flapping limits, respectively.

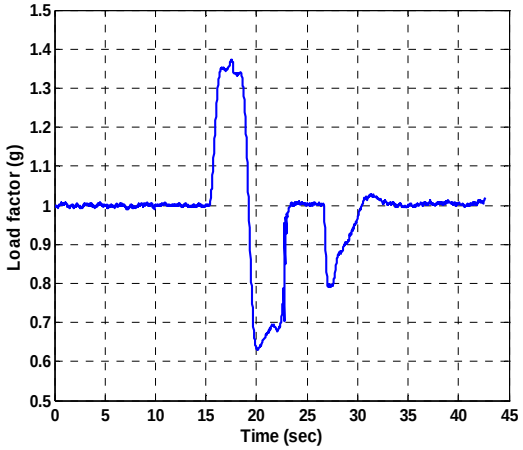


Figure 94. Load factor response for the case of obstacle avoidance with $\pm 3^0$ of flapping limits.

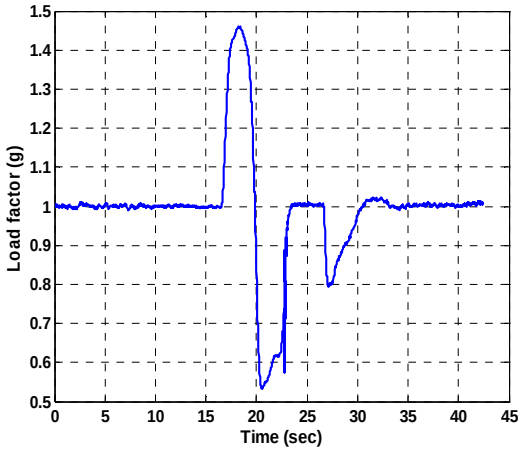


Figure 95. Load factor response for the case of obstacle avoidance with $\pm 5^0$ of flapping limits.

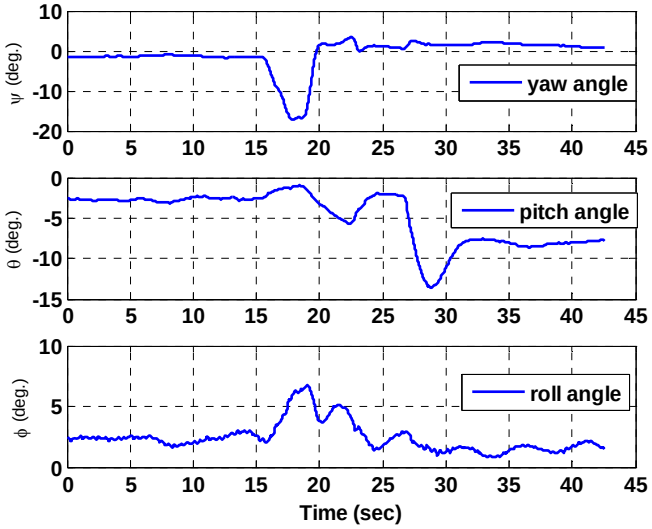


Figure 96. Body attitude responses for the case of obstacle avoidance with $\pm 3^0$ of flapping limits.

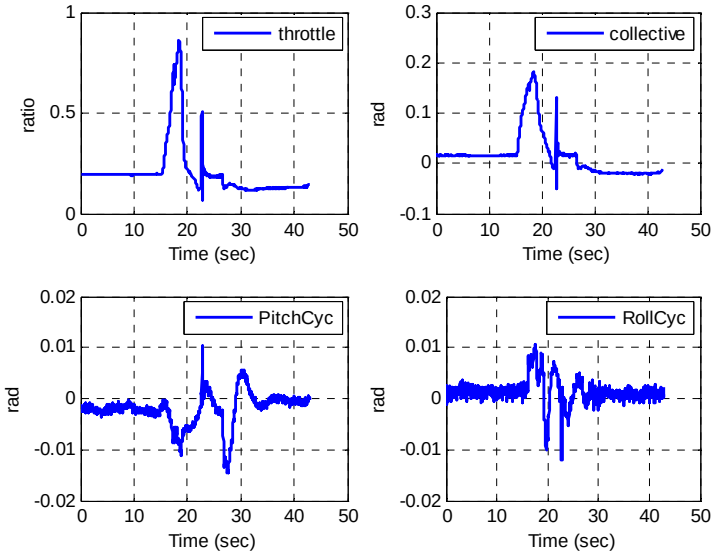


Figure 97. Control responses for the case of obstacle avoidance with $\pm 3^0$ of flapping limits.

4.6 Chapter Summary

In this chapter, an integrated approach is developed to resolve the conflict between aggressive maneuvering needed for obstacle avoidance and the constrained maneuvering needed for envelope protection. A time-optimal solution with obstacle and envelope constraints is used for an integrated approach for obstacle avoidance and envelope protection. According to the relative degree of the limit parameter, the constraint on the limit parameter can be treated as control variable inequality constraint (CVIC) or a state variable inequality constraint (SVIC) by augmenting the state vector with the limit parameter and its derivatives as additional states. The Nonlinear trajectory generator (NTG) is used as a real-time optimization solver.

The computational complexity arising from the obstacle constraints in the NTG solution is reduced by converting the obstacle constraints into a terminal safe waypoint constraint along with an implicit requirement that the horizontal velocity during the avoidance maneuver must be non-negative. The issue of when to initiate a time-optimal avoidance maneuver while staying within the constraints of the vehicle limit boundaries is addressed by including a requirement that the vehicle must maintain its original flight path to the maximum extent possible.

Several case studies presented in this chapter include nominal case, unsafe avoidance case, multiple safe waypoint case, and unidentified obstacle size case. In addition, cases for different limit values and different initial conditions are presented.

The proposed approach is evaluated in simulations using a nonlinear simulation model of a rotary wing UAV test bed within the Georgia Tech UAV simulation tool (GUST). The GUST simulation results presented demonstrate the feasibility of the proposed integrated approach for obstacle avoidance with envelope protection.

CHAPTER 5

THESIS CONTRIBUTIONS, CONCLUDING REMARKS AND RECOMMENDED FUTURE RESEARCH

5.1 Contributions and Conclusions

In this thesis, the design of the guidance systems for an unmanned aerial vehicle (UAV) to execute a given mission is studied. The objective of the guidance system is to determine necessary commands required to achieve mission completion and to send them to a lower level flight controller. Since the way to determine guidance commands is closely connected with which information is available from the sensors and the state estimators, the guidance systems are heavily dependent on situation awareness as well as missions. This thesis does not focus on the development of algorithms to extract the required information from raw measurements of sensing devices and instead assumes that the detection mechanism can be replaced by a simple representation using the detection range and the field of view. The missions considered in this thesis include autonomous formation flight (AFF) in a two-dimensional horizontal plane and obstacle avoidance without violating the safe operational boundaries.

During the development of the formation flight guidance system, it is assumed that a follower UAV is asked to track a maneuvering leader while maintaining the current altitude. Thus, a leader can be a piloted plane, another UAV, or even a ground vehicle. It is also assumed that available information on the relative motion between the leader and the follower is the range and the line-of-sight (LOS) angle. This is based on the fact that the leader is a non-cooperating (non-communicating) vehicle and the follower is

equipped with a passive detection sensor. Thus, it is not possible that time derivatives of the range and the LOS angle are directly measured.

In this thesis, an integrated approach to avoid obstacle collisions and limit violations simultaneously is also studied. While obstacle avoidance is treated as an external constraint, envelope protection is considered as an internal constraint. It is assumed that an obstacle detection device provides information on the size and location of an obstacle according to the detection range and the field of view. It is also assumed that a UAV is asked to minimize the interruption of the original flight path.

The contributions and conclusions are summarized as follows.

5.1.1 Adaptive Guidance Design for Autonomous Formation Flight

In Chapter 3, an adaptive guidance law is designed for a follower UAV to maintaining a relative position with respect to a maneuvering leader vehicle using the theory for multi-input multi-output adaptive output feedback control. The guidance system assumes that the true values of range and LOS angle are available for feedback. It is also assumed that the formation is constrained to remain in the horizontal plane. Based on the fact that the control law design for the formation flight problem exhibits a two time-scale feature, the overall architecture of the autonomous formation flight control system consists of a guidance system and an autopilot system. This implies that the design of the adaptive guidance law is independent of the overall system architecture. Two adaptive guidance laws for AFF are developed and presented: an adaptive velocity command guidance law and an adaptive acceleration command guidance law.

Neural network (NN) based online adaptation is included in the proposed guidance law in order to compensate for the effect of unknown leader maneuvers in the relative distance kinematics and in the LOS angle kinematics. The guidance system determines a velocity command or an acceleration command in order to maintain the relative position as commanded. The adaptive guidance design has the practical

advantage of being simpler to analyze and implement in a situation where there is an existing autopilot system. In the thesis, the adaptive trajectory following controller of the GTMax has been utilized for the overall integration. A detailed description of the GTMax integrated simulation and flight test architecture is presented. The proposed adaptive guidance laws are evaluated using the software-in-the-loop (SILT) simulations assuming that the GTMax is a follower aircraft. Finally, the NN based adaptive velocity command guidance system is implemented within the GTMax rotary wing test bed, and the combined system is evaluated using flight tests.

1. It is seen from the SILT simulations using the velocity command guidance system that with the adaptive component of the guidance law switched off, a steady-state error in the range tracking performance appears whose magnitude is proportional to the speed of the leader. On the other hand, with the adaptive NN switched on, the tracking error has been significantly reduced for the box-shaped maneuver and the racetrack maneuver. This is caused by the fact that the adaptive NN effectively compensates for the effect of the leader velocity along the LOS.

2. Transient response characteristics are compared for the sudden stop maneuver using the velocity guidance command system with and without the NN adaptation. It has been found that the range response without the NN adaptation shows larger transient changes compared to the case with the NN adaptation.

3. Both the adaptive velocity command guidance system and the adaptive acceleration command guidance system are integrated with the existing adaptive trajectory following autopilot controller of the GTMax and compared for the box-shaped maneuver. Performance degradation is seen to occur for the turning maneuver when the adaptive acceleration command system is used. This is caused by the fact that acceleration commands from the guidance system are integrated twice to get position commands required by the adaptive trajectory following controller.

4. The adaptive velocity command guidance system has been evaluated for different range commands: 100 *ft*, 50 *ft*, and 25 *ft*. It is seen from the SITL simulations that the range tracking response shows a significant loss in the performance during the turning maneuvers when the range command is reduced. It can be explained that the leader turning maneuvers for the case of a smaller range command result in rapid changes of the range kinematics.
5. The adaptive velocity command guidance system has been integrated with the GTMax autopilot modified by closing the position feedback loop. The SITL simulation results show this combination has a steady-state error for the box-shaped maneuver.
6. Flight test results using the GTMax and a ground vehicle, respectively, as a follower and a leader show that the NN based adaptive guidance law improves the tracking performance compared to the guidance law without the NN adaptation. This is mainly because the output of the NN can compensate for unknown leader maneuvers.

5.1.2 Integrated Obstacle Avoidance with Envelope Protection

In Chapter 4, an integrated approach to consider obstacle avoidance and envelope protection simultaneously is proposed. It can be easily expected that there will be a conflict between aggressive maneuvering needed for obstacle avoidance and a constrained maneuvering needed for envelope protection. A minimum-time optimal control formulation with obstacle and envelope constraints is used in the proposed approach for obstacle avoidance and envelope protection. An obstacle detection device with certain values of the detection range and the field of view is assumed to identify the obstacle size and location.

In order to reduce the computational complexity from the obstacle avoidance constraints, the original obstacle constraints are converted into a safe waypoint constraint along with an implicit requirement that the horizontal velocity during the avoidance maneuver must be non-negative. A safe waypoint is selected by adding some clearance distance to obstacle configurations. In addition, the way to treat the envelope protection depends on relative degree of the limit parameter. If relative degree of the limit parameter is equal to zero, the constraint on the limit parameter can be considered as a control variable inequality constraint. On the other hand, if relative degree of the limit parameter is greater than or equal to 1, the constraint on the limit parameter can be treated as a state variable inequality constraint (SVIC) by augmenting the state vector with the limit parameter and its derivatives as additional states. The Nonlinear trajectory generator (NTG) is used as a real-time optimization solver.

Once a time-optimal solution is obtained, safety states of the vehicle are determined based on the sign of the optimal horizontal acceleration commands and the sign of the optimal horizontal velocity profile. The safety states are categorized as warning state, safe avoidance state, and unsafe avoidance state. The issue of when to initiate avoidance maneuvers while staying within the constraints of the vehicle limit boundaries is addressed by including a requirement that the vehicle must maintain its original flight path to the maximum extent possible. The proposed method is evaluated in simulations using a nonlinear simulation model of a rotary wing UAV. Artificial values for the load factor limit and the longitudinal flap angle limit are imposed as safe operational boundaries.

1. The proposed algorithm has been tested for the nominal case. It is shown that a UAV maintains the original flight path until the optimal horizontal acceleration command becomes zero if an obstacle is detected early enough for the algorithm to determine the safety state as the warning state. It is shown that with a higher load factor limit value, a UAV can start avoidance maneuver later. In addition, it

is observed that a vehicle needs to start avoidance maneuver earlier if it is originally flying faster.

2. In cases where an obstacle is detected too late to result in an unsafe avoidance situation, the proposed algorithm automatically initiates a deceleration maneuver to reduce the vehicle speed to the point when a safe time-optimal avoidance solution becomes feasible (strategy A) or to the point when the horizontal acceleration command for the time-optimal solution becomes zero (strategy B). It is shown that with strategy A, it takes shorter to reach the safe waypoint from the point of the obstacle detection than with strategy B. However, with strategy B, the vehicle continues along the original path a little loner.

3. It is shown that the proposed algorithm can handle the situations with multiple safe waypoints in a three-dimensional space. The proposed algorithm automatically chooses the best safe waypoint that will result in maintaining the original flight condition to the maximum extent prior to initiating an obstacle avoidance maneuver.

4. For the case of unidentified obstacle size due to the performance limitation of the obstacle detection device, it is noticed that by changing the location of the safe waypoint according to the information obtained up to the current instance, the vehicle can avoid the obstacle while maintaining the limit parameter within the safe boundary.

5. It is shown that a limit on the longitudinal flapping angle can be handled for the obstacle avoidance. Since the original dynamics of the longitudinal flapping angle is not well suited for the NTG application, an approximated dynamic model of the longitudinal flapping angle is used by replacing the pitch rate with the ratio of the vertical acceleration and the speed. It is also noticed that the vehicle maneuvers more aggressively with $\pm 5^o$ flapping angle limit compared to the case with $\pm 3^o$ flapping angle limit.

6. It is observed that neglecting the vehicle plus the autopilot system dynamics results in the degradation of the obstacle avoidance maneuver. Even though the avoidance maneuver is initiated when the horizontal acceleration command becomes zero, the vehicle needs to decelerate until it reaches the safe waypoint. By approximating the vehicle plus the autopilot system dynamics with a first-order dynamics and by selecting appropriate time constant values, the performance of the obstacle avoidance maneuver is improved.

5.2 Recommended Future Research

Several recommended future works related to the research topics studied in this thesis are provided in this section.

5.2.1 Development of multiple-missions control systems

Throughout this thesis, it is assumed that a single mission such as formation flight, obstacle avoidance, etc. is given to a UAV. Each guidance system developed in this thesis is designed to handle a specific mission. However, it can be easily expected that an autonomous controller of UAVs has to manage multiple missions at the same time. For example, a vehicle can encounter an unexpected obstacle, or has to reach given waypoints while flying in formation [48]. For those situations, desired commands to achieve one mission can conflict with other commands. Thus, there is a need to develop multiple-mission control systems for autonomous UAVs in order to manage conflicting maneuvers.

5.2.2 Optimal Guidance Design using the Concept of Hard Limit and Soft Limit

In Chapter 4, an integrated approach for obstacle avoidance with envelope protection using a time-optimal control problem formulation is developed. A safe operational envelope is modeled as constant upper and lower bounds of the limit parameter. In reality, though, there are two types of bounds of limit parameters: One is a 'hard limit bound,' which is never allowed for the limit parameter to hit. The other is a 'soft limit bound,' which is allowed for the limit parameter to stay beyond for short duration. According to this definition, bounds of limit parameters considered in the thesis are hard limit bounds. When different types of bounds of the limit parameters are simultaneously considered, solutions of the time-optimal problem will change, and resulting decision on when to start avoidance maneuver will also change.

One possible alternative is to use different limit bounds according to safety states. When a UAV detects an obstacle along its flight path, a guidance system uses a minimum-time approach with a soft limit bound in order to determine safety states. If the safety state is determined as the warning state or the safe avoidance state, the vehicle maneuvers as developed in this thesis. If the safety state turns out to be the unsafe avoidance state, an optimal solution is reformulated to minimize the limit parameter's excursion beyond the soft limit bound while avoiding an obstacle and maintaining the value of the limit parameter under the hard limit bound.

5.2.3 Integration with Realistic Model of Detection Devices

In this thesis, the problem of how to decide vehicle maneuvers which are required to achieve a given mission is studied. It is assumed that information obtained from detection devices does not include any error due to time-delay, inaccuracy, or bias. In addition, a mechanism of a detection device is simply modeled using the detection range and the field of view. In fact, there are several options for detection devices such as radar,

vision sensor, sonar, etc. Since different types of detection devices have different mechanism, a more realistic model of the detection device is required for evaluation of overall systems. As mentioned in Reference [84], the stability and performance of guidance and control laws are significantly coupled with the performance of sensor systems. Thus, there is a need to investigate integration with realistic model of detection devices.

5.2.4 Investigation of Non-Differentially Flat Limit Parameters

In Chapter 4, the nonlinear trajectory generator (NTG) has been used to solve an optimal control problem in real-time. As mentioned before, the NTG uses differential flatness in order to represent the system in a lower dimension and to increase computational efficiency. However, if the system is not differentially flat, the NTG can not be used for a real-time optimization solver. In general, it is not always possible that dynamics of limit parameters can be expressed or approximated by a function of translational motion variables. Thus, in order to investigate cases with more general limit parameters, a different real-time optimization solver has to be implemented. One may consider DIDO [81] recently developed by Ross as a real-time optimization solver.

5.2.5 Experimental Validation of Integrated Avoidance Algorithm

Integrated avoidance algorithm for both obstacle and limit violation developed in Chapter 4 has been evaluated using the GUST real-time simulation software. Immediate future research recommended is flight test validation of the overall design.

APPENDIX A

DIFFERENTIAL FLATNESS OF SYSTEMS

Definition A.1 The nonlinear system in equation 9 with states $x \in \mathbf{R^n}$ is differentially flat, if there exists a change of variables $z \in \mathbf{R^m}$, given by an equation of the form

$$z = h\left(x, u, \dot{u}, \cdots, u^{(p)}\right) \tag{A.1}$$

such that the states and inputs may be determined from equations of the form

$$(x, u) = g\left(z, \dot{z}, \cdots, z^{(q)}\right) \tag{A.2}$$

The change of variable will transform the system in equation 9 into the trivial system $\dot{z} = v$. Differential flatness is not bound to equilibrium. The transformation may take place around arbitrary trajectories. We will refer to the change of variables z as the flat output. The flat outputs are not necessarily the sensor output of a system.

The significance of a system being flat is that all system behavior can be expressed without integration by the flat outputs and a finite number of its derivatives. That is the problem of finding curves that takes the system from initial conditions to final conditions is reduced to finding any sufficiently smooth curve that satisfies $z^k(0)$ and $z^k(T)$ up to some finite number. There is no need to solve a two-point boundary value problem if the system is differentially flat.

Once all constraints and boundary conditions are mapped into the flat output space, optimal trajectory will be planned in the flat output space and then lifted back to the original state and input space as shown in Figure A1. The idea is that this methodology will alleviate adjoining the system dynamics in the optimal control problem

formulation. Consequently, the number of variables in the optimal control problem will be reduced to expedite real-time computation.

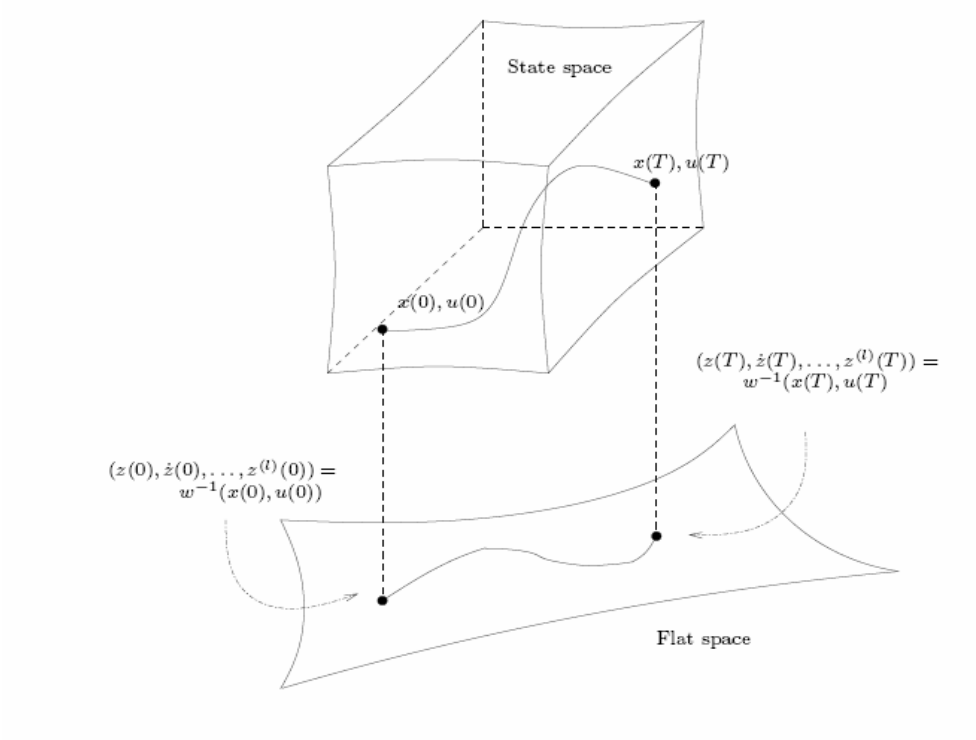


Figure A 1. Mapping from the original system variables to flat outputs

Example The forced VanDerPol oscillator is considered here.

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -x_1(t) + (1 - x_1^{\,2}(t))x_2(t) + u(t) \end{aligned} \tag{A.1}$$

It can be noticed that the forced VanDerPol oscillator is differentially flat with the output $z(t) = x_1(t)$. Then other variables are expressed by

$$\begin{aligned} x_2(t) &= \dot{z}(t) \\ u(t) &= \ddot{z}(t) + z(t) - (1 - z^2(t))\dot{z}(t) \end{aligned} \tag{A.2}$$

REFERENCES

[1] Anderson, M. R. and Robbins, A. C., "Formation Flight as a Cooperative Game," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Reston, VA, August 1998.

[2] Ariyur, K. B., Lommel, P., and Enns, D. F., "Reactive Inflight Obstacle Avoidance via Radar Feedback," *Proceedings of American Control Conference*, Portland, OR, USA, 2005.

[3] Betser, A., Vela, P., and Tannenbaum, A., "Automatic Tracking of Flying Vehicles using Geodesic Snakes and Kalman Filtering," *Proceedings of IEEE Conference on Decision and Control*, Vol. 2, pp. 1649-1654, December 2004.

[4] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, pp. 193-207, 1998.

[5] Bhandari, S. and Colgren, R., "6-DoF Dynamic Model for a Raptor 50 UAV Helicopter Including Stabilizer Bar Dynamics," AIAA 2006-6738.

[6] Bone, E., and Bolkcom, C., *Unmanned Aerial Vehicles: Background and Issues*, Novinka Books, 2004.

[7] Boor, C., *A Practical Guide to Splines*, Springer, 2001.

[8] Borenstein, J. and Koren, Y., "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots," *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 3, pp. 278-288, June 1991.

[9] Bryson, A. E. and Ho. Y., *Applied Optimal Control*, A Halsted Press Book, 1975.

[10] Bryson, A. E., Denham, W. F., and Dreyfus, S. E., "Optimal Programming Problems with Inequality Constraints I: Necessary Conditions for Extremal Solutions," *AIAA Journal*, Vol. 1, pp. 2544-2550, 1963.

[11] Chakravarthy, A., and Ghose, D., "Obstacle Avoidance in a Dynamic Environment: A Collision Cone Approach," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, Vol. 28, 1998.

[12] Charlet, B., Levine, J., and Mario, R., "On Dynamic Feedback Linearization," *Systems and Control Letters*, Vol.13, pp. 143-151, 1989.

[13] Chen, Y., and Huang, J., "A New Computational Approach to Solving a Class of Optimal Control Problems," *International Journal of Control*, Vol.58, pp.1361-1383, 1993.

[14] Cheng, H. L., and Lam, T., "Automatic Guidance and Control for Helicopter Obstacle Avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 6, pp. 1252-1259, 1994.

[15] Clough, B. T., "Unmanned Aerial Vehicles: Autonomous Control Challenges, a Researcher's Perspective," *Journal of Aerospace Computing, Information, and Communication*, Vol. 2, pp. 327-347, 2005.

[16] Cybenko, G., "Approximation by Superposition of a Sigmoidal Function," *Mathematics of Control, Signals, and Systems*, Vol. 2, pp. 303-314, 1989.

[17] Dittrich, J. S., Adolf, F., Langer, A., and Thielecke, F., "Mission Planning for Low-Flying Unmanned Rotorcraft in Unknown Environments," *Proceeding of AHS UAV Specialist Meeting*, Phoenix, AZ, 2007.

[18] Earl, M. G. and Andrea, R., "Iterative MILP Methods for Vehicle-Control Problems," *IEEE Transactions on Robotics*, Vol. 21, pp. 1158-1167, 2005.

[19] Elnagar, G., Kazemi, M. A., and Razzaghi, M., "The Pseudospectral Legendre Method for Discretizing Optimal Control Problems," *IEEE Transactions on Automatic Control*, Vol. 40, pp. 1793-1796, 1995.

[20] Faiz, N., Agrawal, S. K., and Murray, R. M., "Trajectory Planning of Differentially Flat Systems with Dynamics and Inequalities," *Journal of Guidance, Control, and Dynamics*, Vol. 24, pp. 219-227, 2001.

[21] Fliess, M,. Levine, J, Martin, P., and Rouchon, P., "Flatness and Defect of Nonlinear Systmes: Introductory Theory and Examples," *International Journal of Control*, Vol. 61, pp. 1327-1360, 1995.

[22] Floudas, C. A., *Nonlinear and Mixed-Integer Programming: Fundamentals and Applications*, Oxford University Press, 1995.

[23] Geyer, M. S., and Johnson, E. N., "3D Obstacle Avoidance in Adversarial Environments for Unmanned Aerial Vehicles," *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, USA, 2006.

[24] Gill, P., Murray, W., Saunders, M., and Wright, M., *User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming, Systems Optimization Laboratory*, Stanford University.

[25] Gill, P., Murray, W., Saunders, M., and Wright, M., *User's Guide for SNOPT 5.3: A Fortran Package for Large-Scale Nonlinear Programming*, December 1998. University.

[26] Ha, J., Alvino, C., Pryor, G., Niethammer, M., Johnson, E. N., and Tannenbaum, A., "Active Contours and Optical Flow for Automatic Tracking of Flying Vehicles," *Proceedings of American Control Conference*, Vol. 4, pp. 3441-3446, 2004.

[27] Han, S., and Bang, H., "Proportional Navigation-Based Optimal Collision Avoidance for UAVs," *Proceedings of 2nd International Conference on Autonomous Robots and Agents*, 2004.

[28] Hargraves, C. and Paris, S., "Direct Trajectory Optimization using Nonlinear Programming and collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, pp. 338-342.

[29] Horn, J. and Sahani, N., "Detection and Avoidance of Main Rotor Hub Moment Limits on Rotorcraft," *Journal of Aircraft*, Vol. 41, pp. 372-379, 2004.

[30] Horn, J., Calise, A. J., and Prasad, J. V. R., "Flight Envelope Limit Detection and Avoidance for Rotorcraft," *Journal of the American Helicopter Society*, Vol. 47, pp.253-262, 2002.

[31] Hornik, K., Stinchcombe, M., and White, H., "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, Vol. 2, pp. 359-366, 1989.

[32] Hovakimyan, N., Calise, A. J., and Kim, N., "Adaptive Output Feedback Control of a Class of Multi-Input Multi-Output Systems using Neural Networks," *International Journal of Control*, Vol. 77, No. 15, 2004, pp. 1318-1329.

[33] Hovakimyan, N., Nadri, F., Calise, A. J., and Kim, N., "Adaptive Output Feedback Control of Uncertain Nonlinear Systems using Single-Hidden-Layer Neural Networks," *IEEE Transactions on Neural Networks*, Vol.13, 2002, pp. 1420-1431.

[34] Howitt, J., "Carefree Maneuvering in Helicopter Flight Control," *Proceedings of the American Helicopter Society 51st Forum*, Fort Worth, TX, USA, 1995.

[35] Howlett, J., Whalley, M., Tsenkov, P., Schulein, G, and Takahashi, M., "Flight Evaluation of a System for Unmanned Rotorcraft Reactive Navigation in Uncertain Urban Environments," *Proceedings of AHS 63rd Annual Forum*, Virginia Beach, Virginia, 2007.

[36] Hull, D. G., *Optimal Control Theory for Applications*, Springer, 2003.

[37] Inanc, T., Misovec, K., and Murray, R. M., "Nonlinear Trajectory Generation for Unmanned Air Vehicles with Multiple Radars," *Proceedings of IEEE Conference on Decision and Control*, 2004.

[38] Jeram, G., "Open Design for Helicopter Active Control Systems," *Proceedings of the American Helicopter Society 58th Forum*, Montreal, Canada, 2002.

[39] Johnson, E. N., and Calise, A. J., "Limited Authority Adaptive Control of Reusable Launch Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, 2003, pp.906-913.

[40] Johnson, E. N., and Kannan, S., "Adaptive Trajectory Control for Autonomous Helicopters," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 3, 2005, pp.524-538.

[41] Johnson, E. N., and Schrage, D., "The Georgia Tech Unmanned Aerial Research Vehicle: GTMax," AIAA paper 2003-5741, Aug. 2003.

[42] Johnson, E. N., Calise, A. J., Sattigeri, R., Watanabe, Y., and Madyastha, V., "Approaches to Vision-based Formation Flight," *Proceedings of IEEE Conference on Decision and Control*, Vol. 2, pp. 1643-1648, December, 2004.

[43] Kaliath, T., *Linear Systems*, Prentice Hall, 1980.

[44] Khalil, H., *Nonlinear Systems*, Prentice Hall, 2002.

[45] Khatib, O., "Real-Time Obstacle Avoidance for Manipulator and Mobile Robot," *International Journal of Robotics Research*, Vol. 5, pp. 90-98, 1986.

[46] Kim, B., Calise, A. J., Sattigeri, R., "Adaptive, Integrated Guidance and Control Design for Line-of-Sight Based Formation Flight," *AIAA paper* 2006-6716.

[47] Kim, J., and Khosla, P., "Real-Time Obstacle Avoidance Using Harmonic Potential Functions," *Proceedings of IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA, 1991.

[48] Kim, S. and Kim, Y., "Optimum Design of Three-dimensional Behavioural Decentralized Controller for UAV Formation Flight," *Engineering Optimization*, Vol. 41, pp. 199-224, 2009.

[49] Kim, Y. and Lewis, F. L., *High Level Feedback Control with Neural Networks*, World Scientific, 1998.

[50] Kimball, D. F., "Recent Tilt Rotor Flight Control Law Innovations," *Journal of the American Helicopter Society*, Vol. 32, pp. 33-42, 1987.

[51] Knoll, A. and Beck, J., "Autonomous Decision-Making Applied onto UAV Formation Flight," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, August 2006.

[52] Koren, Y., and Borenstein, J., "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," *Proceedings of IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA, 1991.

[53] Lavresky, E., Hovakimyan, N., Calise, A. J., and Stepanyan, V., "Adaptive Vertex Seeking Formation Flight Control," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Austin, TX, August 2003.

[54] Lawrance, C., Zhou, J. and Tits. A., User's Guide for CFSQP Version 2.5, Institute of System Research, University of Maryland.

[55] Lewis, F. L. and Symos, V. L., *Optimal Control*, John Wiley & Sons, 1995.

[56] Li, S. M., Boskovic, J. D., and Mehra, R. K., "Globally Stable Automatic Formation Flight Control in Two Dimensions," *AIAA Paper* 2001-4046, Aug. 2001.

[57] Lozano-Perez, T., and Wesley, M. A., "An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles," *Communications of the ACM*, Vol. 22, pp. 560-570, 1979.

[58] Mason, J. C. and Cox, M. G., *Algorithms for Approximation*, Oxford University Press, 1987.

[59] McGee, T. G., and Hedrick, J. K., "Optimal Path Planning with a Kinematic Airplane Model," *Journal of Guidance, Control, and Dynamics*, Vol. 30, pp. 629-633, 2007.

[60] Milam, M. B., Mushambi, K., and Murray, R. M., "A New Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems," *Proceedings of IEEE Conference on Decision and Control*, 2000.

[61] Milam, M. B., *Real-Time Optimal Trajectory Generation for Constrained Dynamical Systems*, Ph. D. thesis, California Institute of Technology, 2003.

[62] Minguez, J. and Montano, L., "Nearness Diagram Navigation: A New Real Time Collision Avoidance Approach," *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, 2000.

[63] Minguez, J., Montano, L., and Khatib, O., "Reactive Collision Avoidance for Navigation with Dynamic Constraints," *Proceedings of IEEE International Conference on Intelligent Robotics and Systems*, Lausanne, Switzerland, 2002.

[64] Moon, J. and Prasad, J. V. R., "Aggressive Maneuvering for Obstacle Avoidance with Envelope Protection for Autonomous UAVs," *Proceedings of International Specialists' Meeting on Uninhabited Rotorcraft Systems*, Phoenix, AZ, Jan. 2009.

[65] Moon, J. and Prasad, J. V. R., "Minimum-time Approach to Obstacle Avoidance Constrained by Envelope Protection for Autonomous UAVs," *Proceedings of AHS 65th Annual Forum*, Grapevine, TX, May 2009.

[66] Moon, J. and Prasad, J. V. R., "Reactive Obstacle Avoidance for Autonomous UAVs: an Envelope Protection-Based Approach," *Proceedings of AHS 64th Annual Forum*, Montreal, Canada, May 2008.

[67] Moon, J., Prasad, J. V. R., and Calise, A. J., "Guidance Law for the Formation Flight of UAVs: An Adaptive Approach," *Proceedings of AHS 62nd Annual Forum*, May 2006.

[68] Moon, J., Sattigeri, R., Prasad, J.V.R., and Calise, A.J., "Adaptive Guidance and Control for Autonomous UAVs," *Proceedings of AHS 63rd Annual Forum*, May 2007.

[69] Narendra, K. S. and Mukhopadhyay, S., "Intelligent Control Using Neural Networks," *IEEE Control System Magazine*, pp.11-18, 1992.

[70] National Research Council, *Uninhabited Air Vehicles-Enabling Science for Military Systems*, National Academic Press, 2000.

[71] Okubo, A., "Dynamical Aspect of Animal Grouping: Swarms, Schools, Flocks, and Herds," *Advances in Biophysics*, Vol. 22, pp. 1-94, 1986.

[72] Petit, N., Milam, M., and Murray, R. M., "Inversion Based Constrained Trajectory Optimization," *Proceedings of 5th IFAC Symposium on Nonlinear Control Systems*, 2001.

[73] Pollini, L., Giuletti, F., and Innocentri, M., "Sensorless Formation Flight," *AIAA Paper* 2001-4356, Aug. 2003.

[74] Prasad, J. V. R., Calise, A. J., Johnson, E. N., Sattigeri, R., and Moon, J., "Flight Demonstration for an Adaptive Guidance Controller for Autonomous Formation Flight," *Proceedings of AHS 64th Annual Forum*, Montreal, Canada, May 2008.

[75] Prasad, J. V. R., Moon, J., and Kim, C., "Obstacle Avoidance with Envelope Protection for Autonomous UAVs," *Proceedings of the International Forum on Rotorcraft Multidisciplinary Technology*, Seoul, Korea, 2007.

[76] Prasad, J.V.R., Yavrucuk, I., and Unnikrishnan, S., "Adaptive Limit Detection and Avoidance for Rotorcraft," *Proceedings of the 28th European Rotorcraft Forum*, Bristol, UK, 2002.

[77] Proud, A., Pachter, M., and D'Azzo, J., "Close Formation Control," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Portland, OR, August 1999.

[78] Reif, J. and Sharir, M., "Motion Planning in the Presence of Moving Obstacles," *Journal of the Association for Computing Machinery*, Vol. 41, pp. 764-790, 1994.

[79] Richard, A. and How, J., "Aircraft Trajectory Planning with Collision Avoidance using Mixed Integer Linear Programming," *Proceedings of American Control Conference, Anchorage*, AK, 2002.

[80] Rohr, S. N., Lind, R. C., Myers, R. J., Bauson, W. A., Kosiak, W. K., and Yen, H., "An Integrated Approach to Automotive Safety Systems," *Proceedings of Society of Automotive Engineers 2000 World Congress*, Detroit, MI, USA, 2000.

[81] Ross, I. M., *A Beginner's Guide to DIDO*, Elissar, LLC., 2001.

[82] Ross, I. M. and Fahroo, F., "Pseudospectral Methods for Optimal Motion Planning of Differential Flat Systems," *IEEE Transactions on Automatic Control*, Vol. 49, pp. 1410-1413, 2004.

[83] Sahani, N., *Envelope Protection systems for Piloted and Unmanned Rotorcraft*, Ph.D. thesis, The Pennsylvania State University, Department of Aerospace Engineering, 2005.

[84] Sattigeri, R., *Adaptive Estimation and Control with Application to Vision-based Autonomous Formation Flight*, Ph. D. thesis, Georgia Institute of Technology, School of Aerospace Engineering, 2007.

[85] Sattigeri, R., Calise, A. J., Kim, B. S., Volyanskyy, K., and Kim, N., "6-DOF Nonlinear Simulation of Vision-based Formation Flight," *AIAA Paper* 2005-6002, August 2005.

[86] Schouwenaars, T., Valenti, M., Feron, E., and How, J., “Implementation and Flight Test Results of MILP-based UAV Guidance,” *Proceedings of IEEE Aerospace Conference*, Big Sky, MT, USA, 2005.

[87] Schumacher, C. J. and Kumar, R., “Adaptive Control of UAVs in a Closed-Formation Flight,” *Proceedings of the American Control Conference*, Chicago, IL, June 2000.

[88] Seanor, B., Campa, G., Gu, Y., Napolitano, M., Rowe, L., and Perhinschi, M.," Formation Flight Test Result for UAV Research Aircraft Models," *AIAA paper* 2004-6251.

[89] Segal, S., Ben-Asher, J. Z., and Weiss, H., "Derivation of Formation-Flight Guidance Laws for Unmanned Air Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp.733-742.

[90] Seshagiri, S. and Khalil, H. K., “Output Feedback Control of Nonlinear Systems using RBF Neural Networks,” *IEEE Transactions on Neural Network*, Vol. 11, pp. 69-79, 2000.

[91] Sharma, M., and Richard N. D., "Adaptive, Integrated Guidance and Control for Missile Interceptors," *AIAA* 2004-4880.

[92] Speyer, J. L. and Bryson, A. E., “Optimal Programming Problems with a Bounded State Space,” *AIAA Journal*, Vol. 6, pp. 1488-1491, 1968.

[93] Sundar, S., and Shiller, Z., “Optimal Obstacle Avoidance Based on the Hamilton-Jacobi-Bellman Equation,” *IEEE Transactions on Robotics and Automation*, Vol. 13, pp. 305-310, 1997.

[94] Sundar, S., and Shiller, Z., “Time-Optimal Obstacle Avoidance,” *Proceedings of IEEE International Conference on Robotics and Automation*, Nagoya, Japan, 1995.

[95] Tahk, M., Park, C., and Ryoo C., "Line of Sight Guidance Laws for Formation Flight," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp.708-716.

[96] U.S. Air Force, *The U.S. Air Force Remotely Piloted Aircraft and Unmanned Aerial Vehicles Strategic Vision*, 2005.

[97] Unnikrishnan, S. and Prasad, J.V.R., "Flight Evaluation of Reactionary Envelope Protection System for UAVs," *Proceedings of the American Helicopter Society 62nd Forum*, Phoenix, AZ, 2006.

[98] Unnikrishnan, S., *Adaptive Envelope Protection Methods for Aircraft*, Ph. D. thesis, Georgia Institute of Technology, School of Aerospace Engineering, 2006.

[99] Valavanis, K. P., *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*, Springer, 2007.

[100] Verma, A., Wu, C. N., and Castelli, V., "Autonomous Command and Control System for UAV formation," *AIAA Paper* 2003-5704, Aug. 2003.

[101] Von Stryk, O, and Bulirsch, R, "Direct and Indirect Methods for Trajectory Optimization," *Annals of Operations Research*, Vol. 37, pp.357-373, 1992.

[102] Warren, C. W., "A Vector Based Approach to Robot Path Planning," *Proceedings of IEEE International Conference on Robotics and Automation*, Sacramento, CA, USA, 1991.

[103] Watanabe, Y., Calise, A. J., and Johnson, E. N., "Vision-Based Obstacle Avoidance for UAVs," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, Hilton Head, SC, USA, 2007.

[104] Watanabe, Y., Madyastha, V., Johnson, E. N., and Calise, A. J., "Vision-Based Approaches to UAV Formation Flight and Obstacle Avoidance," *2nd International Symposium on Innovative Aerial/Space flyer Systems*, 2005.

[105] Wolfe, J. D., Chichka, D. F., and Speyer, J. L., "Decentralized Controller for Unmanned Aerial Vehicle Formation Flight," *Proceedings of AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, July, 1998.

[106] Yavrucuk, I., *Adaptive Limit Margin Detection and Limit Avoidance*, Ph.D. thesis, Georgia Institute of Technology, School of Aerospace Engineering, 2003.

[107] Zelinski, S., Koo, T. J., and Sastry, S., "Optimization-based Formation Reconfiguration Planning for Autonomous Vehicles," *Proceedings of IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

VITA

JONGKI MOON

Jongki Moon was born in Masan, South Korea on October 3, 1975. He started his undergraduate study at Department of Aerospace Engineering, Seoul National University, Seoul, South Korea in 1994. He received his bachelor's degree and master's degree in aerospace engineering from Seoul National University in 1998 and 2000, respectively. He also worked at Korea Naval Research Laboratory until 2003. He started his PhD study at Georgia Institute of Technology in 2003. He was awarded Vertical Flight Foundation scholarship by American Helicopter Society (AHS) in 2006. He also received his master's degree in mathematics in 2009. His research interests include adaptive control, optimal control, helicopter dynamics, and artificial intelligence.